Chapter 5

Ant Colony Optimization for Resource Allocation and Anomaly Detection in Communication Networks

Lucas Hiera Dias Sampaio,
Mateus de Paula Marques, Mário H. A. C. Adaniya,
Taufik Abrão and Paul Jean E. Jeszensky

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/53338

1. Introduction

Due to the paramount importance of (wireless) communication systems and computer networks, in the last decade both resource allocation (RA) and anomaly detection (AD) problems addressed herein have been intensively studied and numerous solutions have been proposed in the specialized literature [1]. The RA specially in wireless multiple access networks and the AD in computer networks are problems of complex nature and an engineering compromise solution has much appeal in terms of practical and effective deployment.

Resource allocation problems in wireless communication networks include power consumption minimization, information rate and network capacity maximization, battery lifetime maximization, energy-efficient and bandwidth-efficient optimal design among others. In computer networks, anomaly detection system (ADS) consists of a set of techniques aiming to detect anomalies in network operation, helping the administrator to decide which action need to be performed in each situation. Anomaly detection is not an easy task and brings together a range of techniques in several areas, such as machine learning, signal processing techniques based on specification techniques, and data mining among others. Generally, for most scenarios of practical interest, these optimization formulations result in non-convex problems, which is hard to solve or even impossible using conventional convex optimization techniques, even after imposing relaxations to deal with RA problems.

In this sense, heuristics have been widely used to solve problems that deterministic optimization methods result in high computational complexity and therefore have no application in real systems. In this context, the ant colony optimization (ACO) algorithm [2] developed in recent years has attracted a lot of interest from so many professionals due to its robustness and great performance in deal with discrete (combinatorial) and continuous optimization problems.

An important challenge for the future wireless communication systems has been how to acquire higher throughput with lower power consumption. Hence, in order to transfer the exponentially rising amount of available data to the user in an acceptable time, following the "Moore's Law", according to which both the processing power of CPUs and the capacity of mass storage devices doubles approximately every 18 months, the transmission rate in cellular network has been risen at the speed of nearly 10 times every 5 years. Meanwhile, the price paid for this enormous growth in data rates and market penetration is a rising power requirement of information and communication technologies (ICT) – although at a substantially lower speed than "Moore's Law" – the energy consumption doubles every 4-5 years [3].

In order to avoid the collapsing of communication systems and networks resources, an increasing interest and intensive researches in both energy and bandwidth efficient designs have mobilized enormous efforts of research's groups around the globe in the last decade. Against this background, the conventional efficient design of wireless networks mainly focuses on system capacity and spectral efficiency (SE). However, energy-efficient design in wireless networks is of paramount importance and is becoming an inevitable trend, since the deployment of multimedia wireless services and requirement of ubiquitous access have increased rapidly, and as a consequence, energy consumption at both the base station and mobile terminals side have experienced enormous increasing.

In order to achieve high throughput with low total transmit power, system resources such as spectrum (subcarriers, bandwidth), transmit power (energy, battery lifetime) and information rate (QoS requirements) in different multiple access wireless communication systems should be efficiently and appropriately allocated to the different active users. The first part of this chapter is dedicated to deal with the energy-efficient and spectral-efficient designs in DS/CDMA wireless communication systems through the appropriate heuristic optimization of energy and information rate resources.

The Internet has brought to our daily life easy and new ways to execute tasks as searching and gathering information, to communicate and spread ideas and others small gestures that are changing our lives. In order to prevent possible failures and loss of performance, the infrastructure providing theses services must be monitored, which unavoidably increases the responsibility and charge of the network administrator. The administrator is assisted by tools such as firewall, proxy, among others, including the anomaly detection system to help prevent abnormal network operation. Usually the anomaly behavior is a sudden increase or decrease into the network traffic. It can be caused by a simple programming error in some software to hardware failure, among many other causes that affect directly the network operation.

In the next sections of this Chapter the ACO methodology will be applied and analyzed regarding two distinct application of communication scenarios: the resource allocation in a direct sequence code division multiple access (DS/CDMA) systems, which is developed in Section 2 and the anomaly detection in computer networks is discussed in the Section 3. The conclusion remarks for both ACO-communication application problems are offered in Section 4.

2. Resource Allocation in Wireless Multiple Access Networks

The optimized resource allocation in wireless multiple access networks, specially the power rate allocation, is a problem of great interest for telecommunications enterprises and users. It is well known that spectrum and power are valuable resources due to their scarcity, the first one is a natural non renewable resource and the second one is limited by the battery and device size. Therefore, proposing new techniques and algorithms that can allocate this resources in a simple[1] and optimized manner is pretty important.

In the last few decades many researchers have been working on this subject aiming to find a simple yet sturdy algorithm for resource allocation in wireless systems. Among many works recently done we enumerate some notorious in the next section.

2.1. Related Work

Among numerous solutions proposed to resource allocation in wireless multiple access networks we enumerate herein some of great importance works: Foschini and Miljanic [4] distributed power control algorithm (DPCA) stands as the main one. When it comes to metaheuristics, in [5] and [6] a genetic algorithm approach was used to propose the genetic algorithm for mobiles equilibrium, providing the joint power-rate control in CDMA multiple access networks. In [7], the particle swarm optimization (PSO) metaheuristic was used in order to establish a low-complexity power control algorithm. Finally, in [8] a power allocation approach was proposed to solve the parallel interference cancelation in multi-user detectors.

Beyond the metaheuristic approaches, the work developed in [9] exploits an algorithm based on the dynamic cost assignment for downlink power allocation in CDMA networks. Besides, [10] addressed the uplink fairness maximization in a CDMA system with multiple processing gains. In [11], the Verhulst population model, firstly developed to describe the biological species growth with restrictions of space and food, was adapted to the distributed power control problem in a DS/CDMA network. It is noteworthy that this work was the first one to propose a Verhulst model adaptation to resource allocation problems in multiple access networks.

Furthermore, in [12] an analytical approach was proposed for the weighted throughput maximization (WTM) problem, namely MAPEL. The algorithm performs the power control in the interference limited wireless networks, i.e., CDMA and MC/CDMA networks, through a monotonically increasing objective function that is not necessarily convex. This function was formulated as a multiplicative linear fractional programming (MLFP) problem, which is a special case of generalized linear fractional programming (GLFP). So, the GLFP problem presented in [12] was used in [13] in order to formulate a non-decreasing objective function as a weighted SNIR's productory.

Finally, this section presents a heuristic approach through ant colony optimization in the continuous domains ($\text{ACO}_{\mathbb{R}}$) applicable to the power and rate allocation problems [13], and is organized as follows: subsection 2.2 describes aspects of the DS/CDMA networks and the power control problem on subsection 2.3 the power control problem and the cost function used with the ACO algorithm are presented; subsection 2.4 deals with the throughput maximization problem and how the ACO algorithm can be applied to solve this optimization

[1] Here, simple is used as a synonym for low computational complexity.

problem, while in subsection 2.5 the ACO algorithm itself is described. Finally, subsection 2.6 introduces the simulations scenarios, the numerical results and conclusions for the first part of this chapter.

2.2. Resource Allocation in DS/CDMA Networks

In DS/CDMA multirate networks, the Bit Error Rate (BER) is usually used as a QoS metric, since it is directly related to the Signal to Noise plus Interference Ratio (SNIR). Thus, the SNIR is associated to the Carrier to Interference Ratio as follows:

$$\gamma_i = \frac{r_c}{r_i} \times \Gamma_i, \qquad i = 1, \ldots, U \tag{1}$$

where γ_i is the i-th user's SNIR, r_c is the chip rate, r_i is the i-th user's information rate, U is the system load and Γ_i is the i-th user's CIR defined as [11], [14]:

$$\Gamma_i = \frac{p_i |g_{ii}|^2}{\sum_{j=1, i\neq j}^{U} p_j |g_{ij}|^2 + \sigma^2}, i = 1, ..., U \tag{2}$$

where p_i is the i-th user's power bounded by p_{max}, U the number of active users on the system, $|g_{ii}|$ the channel gain of the i-th user, $|g_{ij}|$ is the interfering signals gain and σ^2 the Additive White Gaussian Noise (AWGN) at the i-th receiver's input.

$$\mathbf{G}_{\text{upl}} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1U} \\ g_{21} & g_{22} & \cdots & g_{2U} \\ \vdots & \vdots & \ddots & \vdots \\ g_{U1} & g_{U2} & \cdots & g_{UU} \end{bmatrix} \tag{3}$$

where the main diagonal (g_{ii}) shows the i-th user's channel attenuation, while the other values shows the interfering signals gain.

The path loss is inversely proportional to the distance between the mobile unit and the base station; the shadowing is obtained by a log-normal probability distribution random variable, and the multipath fading obtained assuming a Rayleigh probability distribution for cases without line of sight (LOS), and Rice distribution for cases with LOS.

In the DS/CDMA networks with multiple processing gains (MPG), where each user has a different processing gain $F_i > 1$, it is defined as a function of the chip rate:

$$F_i = \frac{r_c}{r_i}, \qquad i = 1, 2, \ldots, U \tag{4}$$

Therefore, from Eq. (1) and (4) follows:

$$\gamma_i = F_i \times \Gamma_i \tag{5}$$

Additionally, the theoretical Shannon channel capacity is defined as [15]:

$$C = W \log(1 + \gamma) \tag{6}$$

where C is the channel capacity in $bits/s$ and γ is the SNIR. It is worthy to note that since this is a theoretical bound a gap can be included, thus, the Shannon equation can be rewritten as [1]:

$$C = W \log(1 + \theta\gamma) \tag{7}$$

where θ is the gap between the theoretical bound and the real information rate. Usually, θ can be defined as [1]:

$$\theta = -\frac{1.5}{\log(5BER)} \tag{8}$$

where BER is the desired bit error rate (BER).

2.3. Power Allocation Problem

The power control objective is to find the minimal transmission power for each user that satisfy its QoS requirement, usually a minimum transmission rate. Since user rate is related to the user SNIR one may use it as a QoS measure. Thus, the power allocation problem may be mathematically stated as:

$$\begin{aligned} \min \quad & \mathbf{p} = [p_1, p_2, \ldots, p_U] \\ \text{s.t.} \quad & \gamma_i \geq \gamma_i^* \\ & 0 \leq p_i \leq p_{\max} \end{aligned} \tag{9}$$

where p_i and γ_i is the ith user power and SNIR, respectively, and γ_i^* is the desired SNIR level.

In order to enable the users to have minimum QoS warranty, the minimum CIR to SNIR relation must be calculated as [16]:

$$\Gamma_{i,\min} = \frac{r_{i,\min}\gamma_i^*}{r_c}, \quad i = 1, ..., U \tag{10}$$

where $\Gamma_{i,\min}$ and $R_{i,\min}$ are the minimum CIR and minimum information rate of each user, respectively, and γ_i^* is the minimum SNIR needed in order to obtain a minimum BER (or QoS) that is acceptable to each user.

This way, the minimum information rate can be mapped in the SNIR through the Shannon's capacity model using the gap introduced in Eq (8):

$$2^{\frac{r_i}{r_c}} = \max\left[1+\theta_i\gamma_i\right] = \max\left[1+\frac{\theta_i F_i \cdot p_i|g_{ii}|^2}{\sum_{i\neq j}^{U} p_j|g_{ij}|^2+\sigma^2}\right] \tag{11}$$

where $2^{\frac{r_i}{r_c}}$ is the normalized information rate for the i-th user, θ_i is the inverse of the gap between the channel's theoretical capacity and the real information rate. Note that for the minimum SNIR γ_i^*, Eq. 11 uses the minimum information rate established by the system, in order to guarantee QoS. Such that one obtain the condition needed for the minimum SNIR to be satisfied, given a minimum information rate:

$$\gamma_i^* = \frac{2^{r_{i,\min}}-1}{\theta_i} \tag{12}$$

Consider the QoS normalized interference matrix **B** [1]:

$$B_{ij} = \begin{cases} 0, & i=j; \\ \frac{\Gamma_{i,\min} g_{ji}}{g_{ii}}, & \text{otherwise.} \end{cases} \tag{13}$$

which $\Gamma_{i,\min}$ can be obtained as follows:

$$\Gamma_{i,\min} = \frac{r_{i,\min}\gamma_i^*}{r_c}, \quad i=1,...,U \tag{14}$$

Now consider the information rate requirements for each user and the QoS normalized noise vector $u=[u_1,u_2,\ldots,u_k]^T$, with elements:

$$u_i = \frac{\Gamma_{i,\min}\sigma_i^2}{g_{ii}} \tag{15}$$

The solution to the power control problem may be analytically obtained solving the following linear system:

$$\mathbf{p}^* = (\mathbf{I}-\mathbf{B})^{-1}\mathbf{u} \tag{16}$$

where $I_{U\times U}$ is the identity matrix. Note that that $(\mathbf{I}-\mathbf{B})$ is invertible only, and only if the maximum eigenvalue of B is smaller than one [17]. Only in this case, the power control problem will present a feasible solution. Nevertheless, due to the limited resources of mobile terminals, the use of this method is not feasible since its computational cost grows prohibitively when the number of users goes beyond some dozens due to a matrix inversion operation. Besides a totally distributed allocation scheme cannot be deployed using this analytical solution. To overcome this issues, this work proposes a metaheuristic approach for the optimum power-rate allocation problems.

In order to use the ACO algorithm to solve the power allocation problem one must map the problem objective into a mathematical function so-called cost function. In [5, 6] a new cost function for power control problem in CDMA systems using genetic algorithms has been proposed. This function was later modified and used with swarm intelligence in [1] to solve the power control problem. Due to the good results obtained in [1] that cost function was used with the ACO algorithm and reproduced hereafter for convenience:

$$\begin{aligned} J_1(\mathbf{p}) = \max \ & \frac{1}{U}\sum_{i=1}^{U} \mathbb{F}_i^{\text{th}} \cdot \left(1 - \frac{p_i}{p_{\max}}\right), \qquad \forall i = 1, 2, \ldots, U \\ \text{s.t.} \quad & \gamma_i \geq \gamma_i^* \\ & 0 \leq p_i \leq p_{\max} \\ & r_i = r_{i,\min} \end{aligned} \tag{17}$$

where the threshold function is defined as $\mathbb{F}_i^{\text{th}} = \begin{cases} 1, \ \gamma_i \geq \gamma_i^* \\ 0, \ \text{otherwise} \end{cases}$

2.4. Weighted Throughput Maximization (WTM) Problem

The increasing information traffic demand due to multimedia services on third generation networks (3G) and beyond, along with the need of telecommunications companies to improve their profits have motivated development on weighted throughput maximization (WTM) problem, which aims to maximize the system throughput, been formulated as:

$$\begin{aligned} \max_{r} \quad & f(\mathbf{p}) \\ \text{s.t.} \quad & r_i \geq r_{i,\min} \\ & 0 \leq p_i \leq p_{\max} \end{aligned} \tag{18}$$

where $f(\mathbf{p})$ is a cost function that describes the behaviour of information rate of each user regarding the allocated transmit power vector $\mathbf{p}$; r_i is the i-th user's information rate, $r_{i,\min}$ the minimum rate needed to ensure QoS for user i, $\mathbf{p}$ is the power vector such that $\mathbf{p} = [p_1, p_2, \ldots, p_U]$, and $p_{\max}$ is the maximum transmission power allowed in the system.

Therefore, we must incorporate the multirate criterion to the WTM problem subject to maximum power allowed per user. From this, the optimization problem is formulated as a special case of *generalized linear fractional programming* (GLFP) [18]. This way, the second RA problem can be described as follows:

$$\begin{aligned} \underline{J_2}(\mathbf{p}) = \max \quad & \prod_{i=1}^{U} \left[\frac{f_i(\mathbf{p})}{h_i(\mathbf{p})}\right]^{v_i} \\ \text{s.t.} \quad & 0 < p_i \leq p_{i,\max}, \\ & \frac{f_i(\mathbf{p})}{h_i(\mathbf{p})} \geq 2^{r_{i,\min}}, \qquad \forall i = 1, \ldots, U \end{aligned} \tag{19}$$

where $2^{r_{i,\min}}$ is the minimum information rate normalized by the bandwidth of the system (r_c) of the i-th link, including null rate restrictions; $v_i > 0$ is the priority of the i-th user to transmit with satisfied QoS requirements, assumed normalized, such that $\sum_{i=1}^{U} v_i = 1$. It is noteworthy that the second restriction in Eq. (19) is easily obtained from Eqs. (11) and (12), where the minimum information rate given can be transformed in the minimum SNIR through the Shannon capacity equation, considering a maximum tolerable BER for each user or service class. Hence, functions $f_i(\mathbf{p})$ and $h_i(\mathbf{p})$ can be readily defined as:

$$h_i(\mathbf{p}) = \sum_{j \neq i}^{U} p_j |g_{ij}|^2 + \sigma^2 \qquad \text{and} \qquad f_i(\mathbf{p}) = \theta F_i \cdot p_i |g_{ii}|^2 + h_i(\mathbf{p}), \qquad \forall i = 1, \ldots, U. \tag{20}$$

Note that the Eq. (19) is the productory of linear fractional exponentiated functions, and the function $\prod_{i=1}^{U} (z_i)^{v_i}$ is an increasing function in a nonnegative real domain [12]. Based on these properties, problem (19) can be properly rewritten as:

$$\begin{aligned} J_2(\mathbf{p}) = \max \sum_{i=1}^{U} v_i [\log_2 f_i(\mathbf{p}) - \log_2 h_i(\mathbf{p})] \quad &= \quad \max \sum_{i=1}^{U} v_i [\bar{f}_i(\mathbf{p}) - \bar{h}_i(\mathbf{p})] \\ \text{s.t.} \quad 0 < p_i \leq p_{i,\max}, & \\ \bar{f}_i(\mathbf{p}) - \bar{h}_i(\mathbf{p}) \geq r_{i,\min}, \quad \forall i = 1, \ldots, U & \end{aligned} \tag{21}$$

This way, the cost function turns into a sum of logarithms, which results in a monotonic nondecreasing function. With no loss of generality, in this work $v_i = U^{-1}, \forall i$, has been adopted.

2.5. The ACO$_{\mathbb{R}}$ Metaheuristic

The ACO$_{\mathbb{R}}$ is a continuous-valued metaheuristic based on the ants behavior when looking for food. Note that it was first proposed for combinatorial optimization problems. In its discrete version, each ant walks through the points of the input set and deposits pheromone on its edges. The next point selection is done probabilistically, considering the amount of pheromone on each edge, jointly with the heuristic information available in the current algorithm iteration.

Given a set of points next to an ant, the probability of each of this points to be chosen forms a probability mass function (PMF). The main idea of the continuous version ACO$_{\mathbb{R}}$ is the adaptation of this PMF to a Probability Density Function (PDF), allowing each ant to sample a continuous PDF instead of dealing with discrete sampling points. This is due to the fact that the continuous domain has infinite points to be chosen.

The PDF used in this work is Gaussian given its soft capacity in generating random numbers, and due to the fact that it has only one maximum point located at the mean of the process. Nevertheless, this last feature is not useful when the search space has more than one feasible region. To overcome this problem, the ACO$_{\mathbb{R}}$ uses a Gaussian kernel pdf (a weighted sum of Gaussians) to sample each dimension of the problem. Each Gaussian kernel is defined as follows [19]:

$$G^i(x) = \sum_{l=1}^{Fs} \omega_l g_l^i(x) = \sum_{l=1}^{Fs} \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} \exp\left[-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i^2}} \right], \qquad i = 1,\dots,U \tag{22}$$

where i is the Gaussian kernel indexer, with U being the number of dimensions of the problem; $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_{Fs}]$ is the weight vector associated to each Gaussian in the kernel; $\boldsymbol{\mu}^i = [\mu_1^i, \mu_2^i, \dots, \mu_{Fs}^i]$ is the vector of means and $\boldsymbol{\sigma}^i = [\sigma_1^i, \sigma_2^i, \dots, \sigma_{Fs}^i]$ is the vector of standard deviations. Hence, the cardinality of both vectors is equal to the number of Gaussians in the set, $|\boldsymbol{\omega}| = |\boldsymbol{\mu}^i| = |\boldsymbol{\sigma}^i| = Fs$.

For discrete combinatorial optimization problems, the pheromone informations are kept in a table. This is not possible when we need to deal with continuous problems, since there are an infinite number of points to keep, and as a consequence, an infinite ways to evolve. Thus, a solution file is deployed, where the lth solution s_l, $\forall l = 1, 2, \dots, Fs$, in the ith dimension, $\forall i = 1, \dots, U$, is kept on the memory file at the nth iteration, as well as the respective cost function values $f(s_l)$. A schematic grid to understand the file format and the associate ACO input parameters is sketched in Fig. 1.

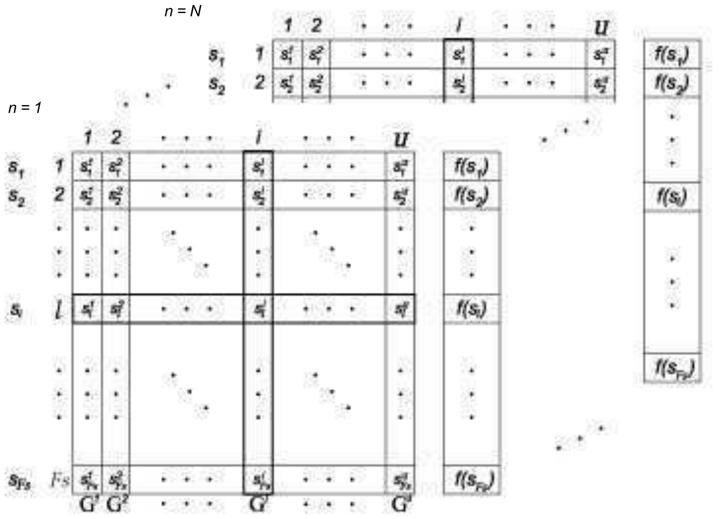

n = N
n = 1
1 2 ... i ... U
s_1 1
s_2 2
s_l l
s_{Fs} Fs
s_1^1 s_1^2 s_1^i s_1^U
s_2^1 s_2^2 s_2^i s_2^U
s_l^1 s_l^2 s_l^i s_l^U
s_{Fs}^1 s_{Fs}^2 s_{Fs}^i s_{Fs}^U
G^1 G^2 ... G^i ... G^U
$f(s_1)$
$f(s_2)$
$f(s_l)$
$f(s_{Fs})$

Figure 1. File structure for the ACO algorithm's solutions. Each line of the $Fs \times U$ matrix represents one solution for the problem of dimension U; each column represents one dimension, which, in turn, is sampled by each Gaussian kernel. Cost function vector $\mathbf{J} = [J(s_1), \dots, J(s_l), \dots, J(s_{Fs})]$, dimension $Fs \times 1$, represents the solution for the n-th iteration. Finally, each layer (in depth) shows the solution file in each iteration from $n = 1$ to N.

The found solutions are used to generate PDFs dynamically, through a method based on the stored solutions. The vectors $\boldsymbol{\mu}^i$ and $\boldsymbol{\sigma}^i$ at ith dimension and $\boldsymbol{\omega}$ common for all dimensions are calculated through the solutions of the file at each algorithm iteration; so, the Gaussian kernel can be built, and guide the ants throughout the dimensions of the problem.

The solutions file must store Fs solutions. Note that this number is equal to the number of Gaussian PDFs ($g_l^i(x)$) in the i-th kernel (G^i). Thus, one can conclude that the i-th Gaussian kernel will have one Gaussian PDF sampling each i-th variable of each solution. Herein, the greater the value of Fs, the greater will be the number of Gaussian PDFs on the algorithm. Therefore, the parameter Fs leads to the complexity of the set.

A detailed description for the evolution of the ACO solution structure is given in the following. From Fig. 1, note that for an U-dimensional problem, the file solution stores Fs different solutions, the values of its U variables and the results of each solution applied to the cost function. So, s_l^i is the value of the i-th variable of the l-th solution of the file, and $J(s_l)$ is the result of the l-th solution applied to the cost function. For each dimension $i = 1, \ldots, U$ of the problem (in this case, each column of the table), there is a different Gaussian Kernel PDF (G^i) defined. So, for each G^i, the values of the i-th variable of all solutions becomes the elements of the mean vector, $\boldsymbol{\mu}^i = [\mu_1^i, \ldots, \mu_{Fs}^i] = [s_1^i, \ldots, s_{Fs}^i]$, i.e., the l-th value of dimension i is the mean of the l-th gaussian of G^i.

Furthermore, the number of ants (or particles) m is another important input parameter of the ACO algorithm to be adjusted. The ants are responsible for the sampling of G^i, and thus, for the algorithm evolution as well. In that way, on each iteration, each ant chooses one solution of the file probabilistically, through a method based on the weight vector $\boldsymbol{\omega}$. Since the ant has chosen one solution of the file, the next step consists of sampling through the Gaussian kernel. After that, a new solution is generated and attached to the end of the file. As the last ant finishes its sampling, the solution file is sorted based on the value entries in the cost function matrix $\mathbf{J} = [J(s_1), \ldots, J(s_l), \ldots, J(s_{Fs})]$. Hence, for both problems treated in Eq. (17) and (21), the matrix $\mathbf{J}$ must be sorted decreasingly, i.e., $J(s_1) \geq J(s_2) \geq \ldots \geq J(s_{Fs})$.

When the sorting process is completed, a number of worst solutions ordered at the end of the file is discarded, in which is done equal to the number of solutions added on the sampling process. Note that since each ant samples only one solution on each iteration, the number of solutions to be discarded is equal to the number of ants.

At this point, a Gaussian kernel (G^i) is defined for each dimension of the problem, which the l-th variable becomes an element of the $\boldsymbol{\mu}^i$ vector. Thus, considering the G^i defined on the i-th dimension, The weight w_l of each solution is calculated as follows:

$$\omega_l = \frac{1}{qFs\sqrt{2\pi}} \exp\left[-\frac{(l-1)^2}{2q^2Fs^2}\right], \qquad l = 1, \ldots, Fs \tag{23}$$

The weight of the l-th solution can be seen as the probability of a solution to be chosen and sampled by an ant. Hence, the l-th solution's rank in the file, also is the input parameter in Eq. (23), which is a Gaussian PDF with mean 1 and standard deviation $q \cdot Fs$, where q is an input parameter of the ACO algorithm. The q parameter can be interpreted as a diversification parameter, where low values of q enhances the convergence speed of the algorithm; on the other hand, high values entries for q enhances the process robustness. This is due to the fact that, on the normal function for the l-th solution's weight calculation in Eq. (23), the higher the standard deviation values are, more chances to select solutions that are not so near to the mean of the process, which, in turn, is the first solution of the file. So, when the standard deviation $q \cdot Fs$ assumes small values, only the best and a few solutions of the file will be sampled, enabling the algorithm to converge faster. On the other hand, when high-valued standard deviations are admitted, the probability of the file solutions to be chosen becomes more uniform, which makes the algorithm search in a larger space (more diversity), at the cost of lower convergence speed. Thus, the ACO$_{\mathbb{R}}$'s q parameter corresponds to the best-so-far solution and iteration-best solution concepts. So, Eq. (23) gives rise to an important equilibrium between q and Fs parameters, making their individual calibration sensitive to each one, in order to achieve a good tradeoff among robustness and convergence velocity for each specific optimization problem.

Furthermore, the suitable choice for the population size m plays an important role in order to improve the robustness-speed tradeoff in conjunction with the best attainable $q \cdot Fs$ calibration. Note that m might be overloaded in order to increase the algorithm capacities, at the undesirable cost of greater computational complexity. Finally, the algorithm robustness $\mathcal{R}$ can be thought as the ratio between the number of convergence success $\mathcal{S}$ to the total number of process realizations $\mathcal{T}$:

$$\mathcal{R} = \frac{\mathcal{S}}{\mathcal{T}} \cdot 100 \qquad [\%] \qquad @N \quad \text{iterations}$$

and the speed as the average number of iterations needed to the algorithm achieves convergence in $\mathcal{T}$ trials for a given problem.

Next, important steps of the ACO algorithm are briefly discussed, such as general ACO algorithm structure, the $\boldsymbol{\sigma}^i$ vector computation, as well as the sampling process; the last one will be presented directly on the algorithm structure. The algorithm organization common to various implemented versions of continuous ACO is described in Algorithm 1, in which the functions performed inside are briefly described in following.

Algorithm 1 Overview of ACO

```
while The end conditions aren't met do
    AntBasedSolutionConstruction()
    PheromoneUpdate()
    DaemonActions()
end while
```

- *AntBasedSolutionConstruction()*: Through the decision variables of each solution $s_l^i, i = 1, \ldots, U$, each ant builds the solution by U steps. Since ACO$_{\mathbb{R}}$ uses a Gaussian mixture in each problem dimension (Eq. (22)), and that the number of Gaussians on the mixture is equal to the size Fs of the solutions file, we conclude that at each step i we will have a different sample of G^i.

  In order to sample the Gaussian kernel, the vectors $\boldsymbol{\mu}^i$, $\boldsymbol{\sigma}^i$ and $\boldsymbol{\omega}$ must be updated. In this work, the vector $\boldsymbol{\omega}$ will not be used, and the explanation for that is given in the next paragraph.

  In practice, the sample process is made on three stages: First, the elements of $\boldsymbol{\omega}$ vector must be calculated, where should be noted that the solutions ranking parameter l will never change, independently of the change of the solutions order on the file. On the second stage, each ant must choose a solution of the file aiming to sample it, and the probability of this choose must be relative to the normalization of each solution weight for the sum of all weights:

$$p_l = \omega_l \cdot \left(\sum_{r=1}^{k} \omega_r \right)^{-1} \tag{24}$$

that is, the probability of each solution being chosen can be thought as a random number generator of normal distribution, with mean 1 and standard deviation $q \times Fs$, since the choose probability of each rank will never change. Adopting this strategy, the ω vector as well as the first stage of the sampling process will no longer be needed.

Thus, since the ant chosen its solution, it must be sampled stepwise using a normal random number generator. The chosen solution must be sampled dimensionally ($g_l^i, i = 1, \ldots, U$), causing each Gaussian mixture's parameters to be seen only in one dimension a time, smoothing the calculation of the pattern deviation and allowing linear transformations on the problem without result changes.

Therefore, let s_l (Fig. 1) to be the solution chosen by an ant during the ACO's evolution process. It is known that the ant will sample s_l dimensionally, as well as that the sampling is done through a Gaussian function parametrized by the μ_l^i and σ_l^i values. Thus, the σ_l^i is calculated for dimension i as follows:

$$\sigma_l^i = \xi \sum_{e=1}^{k} \frac{|s_e^i - s_l^i|}{k-1} \tag{25}$$

Herein, the σ_l^i value for s_l^i is the mean distance from s_l^i to the other values of dimension i in the other solutions of the file. The process is repeated until the last dimension of the file is reached. This way, the higher the variability of the different solutions, higher will be the standard deviation value σ_l^i. Note that the $\xi \in [0, 1]$ parameter aims to reduce the standard deviation, working as a learning factor. Since the σ_l^i value is calculated, the ant will sample the Gaussian PDF $g_l^i(\mu_l^i, \sigma_l^i)$.

The parameter ξ is the same for all dimensions of all solutions, and corresponds to the pheromone evaporation rate, or to the inverse of the learning rate. This way, when ξ is low valued the algorithm speed is enhanced, and when it is high-valued, its robustness is enhanced. It is noteworthy that the algorithm converges when $\sigma \to 0$ throughout all dimensions of the file.

- *PheromoneUpdate()*: The ACO$_{\mathbb{R}}$ algorithm updates its pheromone informations as follows: At the beginning of the algorithm, the file is initialized with Fs solutions uniformly random distributed. From this, the pheromone updating is done adding the new solutions generated by the ants, as well as removing the same number of worst solutions.

  Finally, the size of the solutions file is a parameter of the algorithm, and must not be smaller than the number of dimensions of the problem if it is enabled to handle variable correlation, and to support linear transformations on the problem being optimized. Nevertheless, these techniques are not used in this work. Furthermore, the size of the file leads to the algorithm diversity, since a big file will cover a greater region of the search space than the small one, enabling the algorithm to overcome local optima, but on the other hand, a small file will make the algorithm to be faster than the big one.

- *DaemonActions()*: This is the optional component of ACO$_{\mathbb{R}}$ that can be used to implement centralized actions of the algorithm that aren't accessible for the ants. In this stage, the found solution must be updated and returned as the final solution. Besides, it is possible to implement local search methods here, but this aspect is not exploited in this current work, since we look firstly for low computational complexity.

2.6. Numerical Results

This subsection is divided in two parts. The first one deals with ACO typical performance and its input parameters optimization; the second part numerical simulations results for both power and rate allocation problems are presented and the NMSE is compared with the RA-PSO algorithm.

The simulations were carried out in the MatLab 7.0 platform and the scenario parameters are presented on Table 1. We assumed a rectangular cell with one base station in the center and users uniformly spread across all the cell extension. We considered that all mobile terminals experience slow fading channels, i.e. the following relation is always satisfied:

$$T_{\text{slot}} < (\Delta t)_c \tag{26}$$

where $T_{\text{slot}} = R_{\text{slot}}^{-1}$ is the time slot duration, R_{slot} is the transmitted power vector update rate, and $(\Delta t)_c$ is the coherence time of the channel[2]. This is part of the SNIR estimation process, which means that the channel is constant in each optimization window, assumed herein equal to $667\mu s$. Thus, the ACO algorithm must converge to the solution within each $667\mu s$ interval.

2.6.1. RA-ACO Input Parameters Optimization

Simulation experiments were carried out in order to determine the suitable values for the ACO input parameters for each problem, such as file size (Fs), pheromone evaporation coefficient (ξ), population (m) and the diversity parameter (q). The best parameters combination was chosen considering the solutions quality measured by the normalized mean squared error (NMSE), defined as:

$$\text{NMSE} = \mathbb{E}\left[\frac{||\mathbf{p} - \mathbf{p}^*||^2}{||\mathbf{p}^*||}\right] \tag{27}$$

where $\mathbf{p}$ is the solution found through the ACO algorithm, $\mathbf{p}^*$ the analytical (optimal) solution and $\mathbb{E}$ the mathematical expectation operator. In order to find the best parameters for both problems non-exhaustive tests were conducted.

A typical convergence behavior for the RA-ACO under equal-rate power control problem is shown in Fig. 2. At a first glance, power allocation for $U = 5$ users (lightly loading system) is performed by a) RA-ACO algorithm, b) RA-PSO algorithm from [1]. One can

[2] Corresponds to the time interval in which the channel characteristics do not suffer expressive variations.

Parameters	Adopted Values
DS/CDMA Power-Rate Allocation System	
Noise Power	$P_n = -63$ [dBm]
Chip rate	$r_c = 3.84 \times 10^6$
Min. Signal-noise ratio	$SNR_{\min} = 4$ dB
Max. power per user	$P_{\max} = 1$ [W]
Min. Power per user	$P_{\min} = 0$ [W]
Time slot duration	$T_{\text{slot}} = 666.7\mu$s or $R_{\text{slot}} = 1500$ slots/s
# mobile terminals	$U \in \{5; 10; 20; 100; 250\}$ users
# base station	BS = 1
Cell geometry	rectangular, with $x_{\text{cell}} = y_{\text{cell}} = 5$ Km
Mobile terminals distrib.	$\sim \mathcal{U}[x_{\text{cell}}, y_{\text{cell}}]$
Fading Channel Type	
Path loss	$\propto d^{-2}$
Shadowing	uncorrelated log-normal, $\sigma^2 = 6$ dB
Fading	Rice: [0.6; 0.4]
Time selectivity	slow
User Features and QoS	
User Services	[voice; video; data]
User Rates	$r_{i,\min} = [\frac{r_c}{128}; \frac{r_c}{32}; \frac{r_c}{16}]$ [bps]
User BER	$BER = [5 \times 10^{-3}; 5 \times 10^{-5}; 5 \times 10^{-8}]$
RA-ACO Algorithm	
Problem Dimensionality	$U \in \{5; 10; 20; 100; 250\}$ users
File Size	$Fs \in [8, 25]$
Diversity Factor	$q \in [0, 1]$;
Pheromone Evaporation Rate	$\xi \in [0, 1]$;
Population Size	$m \in [7, 35]$;
Max. # iterations	$N = 1000$
Monte-Carlo Simulation	
Trials number	$\mathcal{T} = 1000$ realizations

Table 1. Multirate DS/CDMA system, channel and ACO input parameters

see the smooth-monotonic convergence of the RA-ACO algorithm toward the optimal power solution, in this case given by (16), in contrast to the non-monotonic oscillated convergence behavior presented by the RA-PSO algorithm. Besides, for $U = 5$ users power allocation problem, the ACO was able to achieve convergence after ≈ 250 iterations in contrast to the ≈ 450 iterations necessary for the RA-PSO convergence.

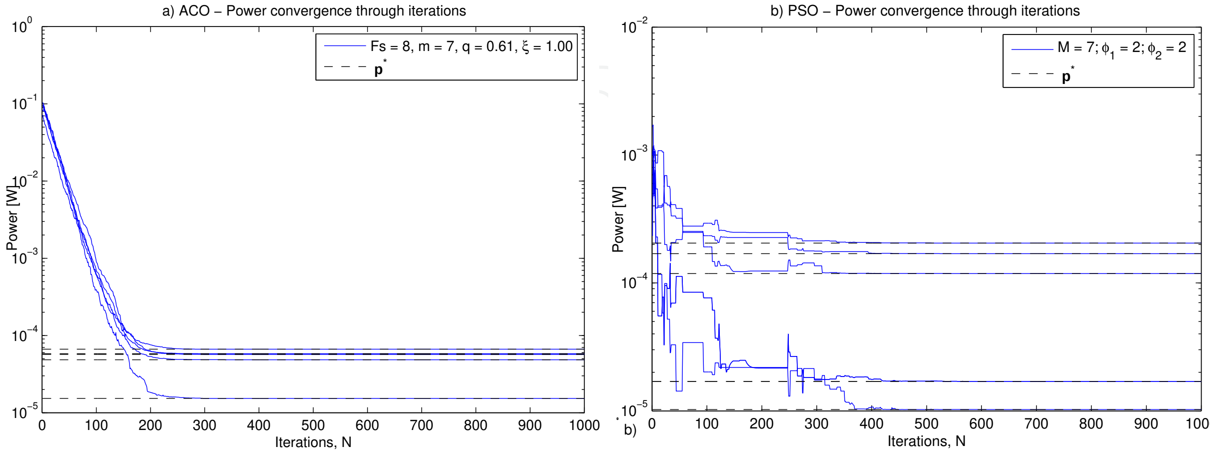


Figure 2. Power allocation for $U = 5$ users. Equally information rate among users is adopted. a) RA-ACO; b) RA-PSO algorithm from [1].

Power Allocation (PA) Problem. Under the first resource allocation problem posed by Eq. (9) or (17), Fig. 3 depicts the associated NMSE under different ACO input parameter values combination taking into account different loading system, i.e., $U = 5$, 10 and 20, respectively.

Note that the population size m and file size Fs parameters ($m, Fs \in \mathbb{N}$), both with entry values common for all the different $\{q, \xi\}$ input parameters configurations, where chosen based on the problem dimensionality. Numerical experiments have shown that different entries around the ones chosen do not affect substantially the NMSE results as the different entries for q and ξ parameters do. It is worth noting that the PA problem in (9) presents a non-convex characteristic; hence, the value entries for the population size m and file size Fs parameters assume relative high values regarding the dimensions of the problem, meaning that both parameters are of the order of problem dimension, $\{m, Fs\} \approx \mathcal{O}[U]$. It means that RA-ACO can solve the non-convex PA problem in DS/CDMA systems but with input parameter loads relatively high.

Herein, a parameter calibration strategy was adopted in order to find the best tradeoff for the $\{q; Fs\}$ set, given in Eq. (23). Since the parameters Fs and m are directly related to the computational complexity of the algorithm, finding a suitable parameter set with Fs entries as low as possible is of great interest.

On the other hand, the population size m parameter has a small or even no influence on the any other ACO input parameter (as q and Fs interfere each other). Although the m entries values directly increases the algorithm computational complexity. Therefore, the parameters m and Fs were fixed at low values and then the best q and ξ combination for it was sought. Hence, based on the NMSE *versus* convergence speed results obtained in Fig. 3, the optimized RA-ACO input q and ξ parameters for the power control problem in DS/CDMA networks under different level of interference could be found, as summarized in Table 2.

U (users)	5	10	20
q	0.61	0.40	0.40
ξ	1.00	0.82	0.75
m	7	15	35
Fs	8	4	25
Robustness, $\mathcal{R}$	100 %	100 %	30 %

Table 2. Optimized RA-ACO input parameters and respective robustness for the Problem of Eq. (17).

Also, the robustness achieved by the RA-ACO for the power allocation problem is added to the Table 2. Herein, the success of convergence is reached when the NMSE of the algorithm's solution goes less than 10^{-2}. Due to the non-convexity of the PA problem in (17), when the number of users grows from 10 to 20, the needed robustnes grows exponentially, thus, the algorithm's performance have a critical decay of 70%.

Weighted Throughput Maximization (WTM) Problem. For the weighted throughput maximization (WTM) problem posed in Eq. (21), Figure 4 shows different cost function evolutions when parameters q and ξ are combined under three distinct system loading, $U = 20$, 100 and 250 users. The average cost function evolution values where taken over $\mathcal{T} = 1000$ trials. Also, the correspondent sum rate difference $(\Delta \sum_{\text{rate}})$ is zoomed in.

From Fig. 4-a it is clear that for $U = 20$ users, the $q = 0.10$ and $\xi = 1.00$ choice results in an average cost function value higher than the other ones. Besides, even in a

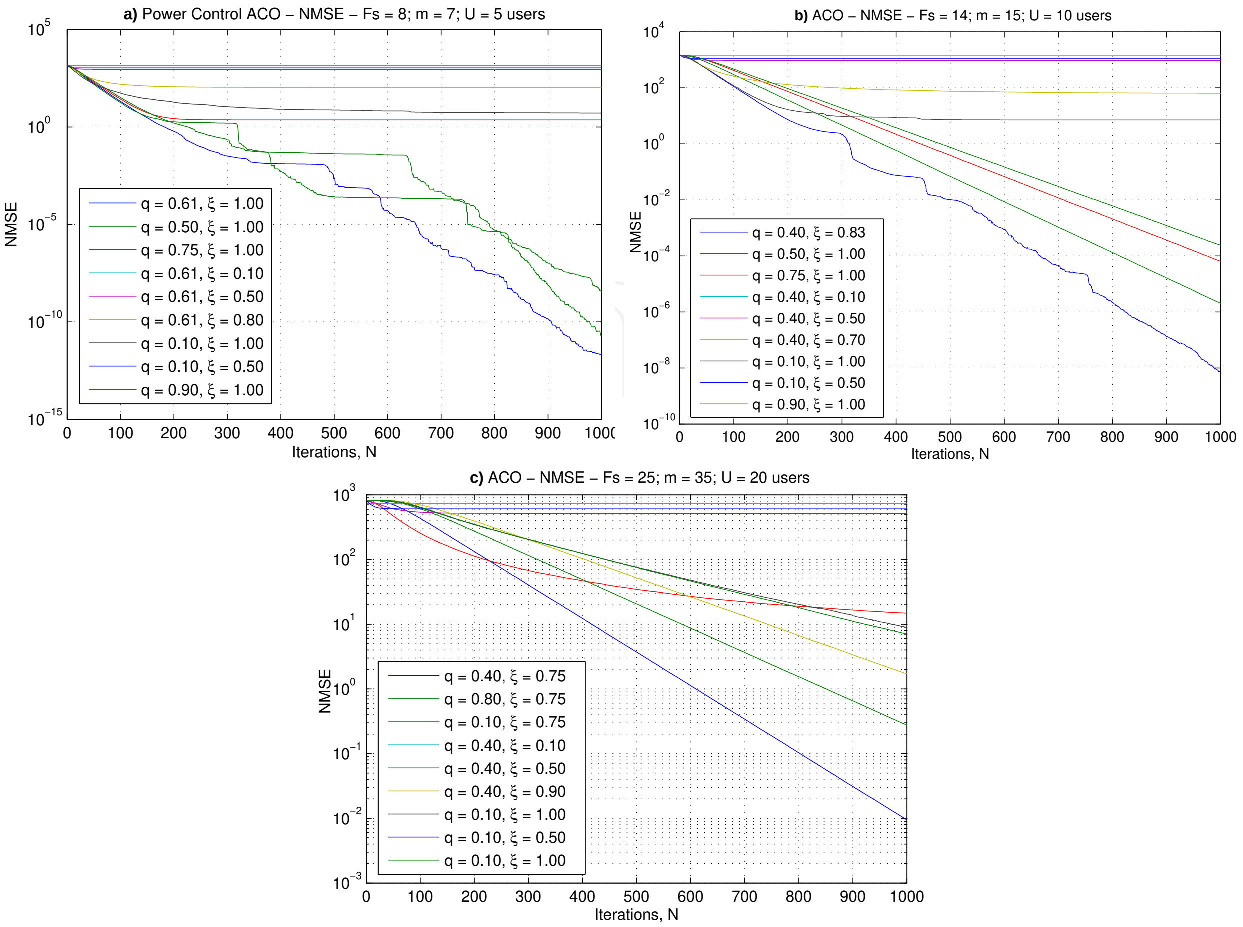


Figure 3. NMSE for different ACO input parameters. a) $U = 5$ users; b) $U = 10$ users and c) $U = 20$ users;

relative course optimization scenarios for q and ξ parameters it is clear the importance of deploying optimized RA-ACO input parameters; for instance, the best ACO input parameters configuration in Figure 4.a shows a difference of $\Delta \sum_{\text{rate}} = 76.8Kb/s$ on the total achievable system throughput regarding the second best parameters set choice, and a difference of $1.3Mb/s$ to the worst parameters set.

On Fig. 4.b, a lightly cost function value difference shows that the best parameter configuration for the system load of $U = 100$ users is $q = 0.20$ and $\xi = 1.00$. In terms of system throughput, the best parameter configuration shows a difference of $38.4Kb/s$ to the second one, and of $2Mb/s$ to the worst one.

Finally, the best input parameters configuration set for $U = 250$ users in Fig. 4.c is obtained as $q = 0.20$ and $\xi = 1.00$. Again, the associated total system throughput variations due to the different ACO input parameter configurations was not significant, ranging from $96Kb/s$ to $1.24Mb/s$. This confirms a certain robustness to the input $\{q; \xi\}$ deviation values, thanks to the convexity of the optimization problem formulated in (21). In summary, the RA-ACO algorithm was able to attain reasonable converge in less than $n = 150$ iterations for the WTM problem with dimension up to $U = 250$ users.

Note that in the WTM problem given the convex characteristic of the objective function, Eq. (21), the robustness of the RA-ACO approaches to $\mathcal{R} \approx 100\%$, and the value entries for the population size m and file size Fs parameters are impressively less than the number of dimensions of the problem, i.e., $\{m, Fs\} \ll U$. It means that RA-ACO can solve the WTM problem with soft parameter loads. The best input parameter configuration for the RA-ACO algorithm in order to solve the WTM problem is summarized in Table 3.

U (users)	20	100	250
q	0.10	0.20	0.20
ξ	1.00	1.00	1.00
m	3	7	7
Fs	5	5	5

Table 3. Optimized RA-ACO input parameters for the WTM Problem, Eq. (21).

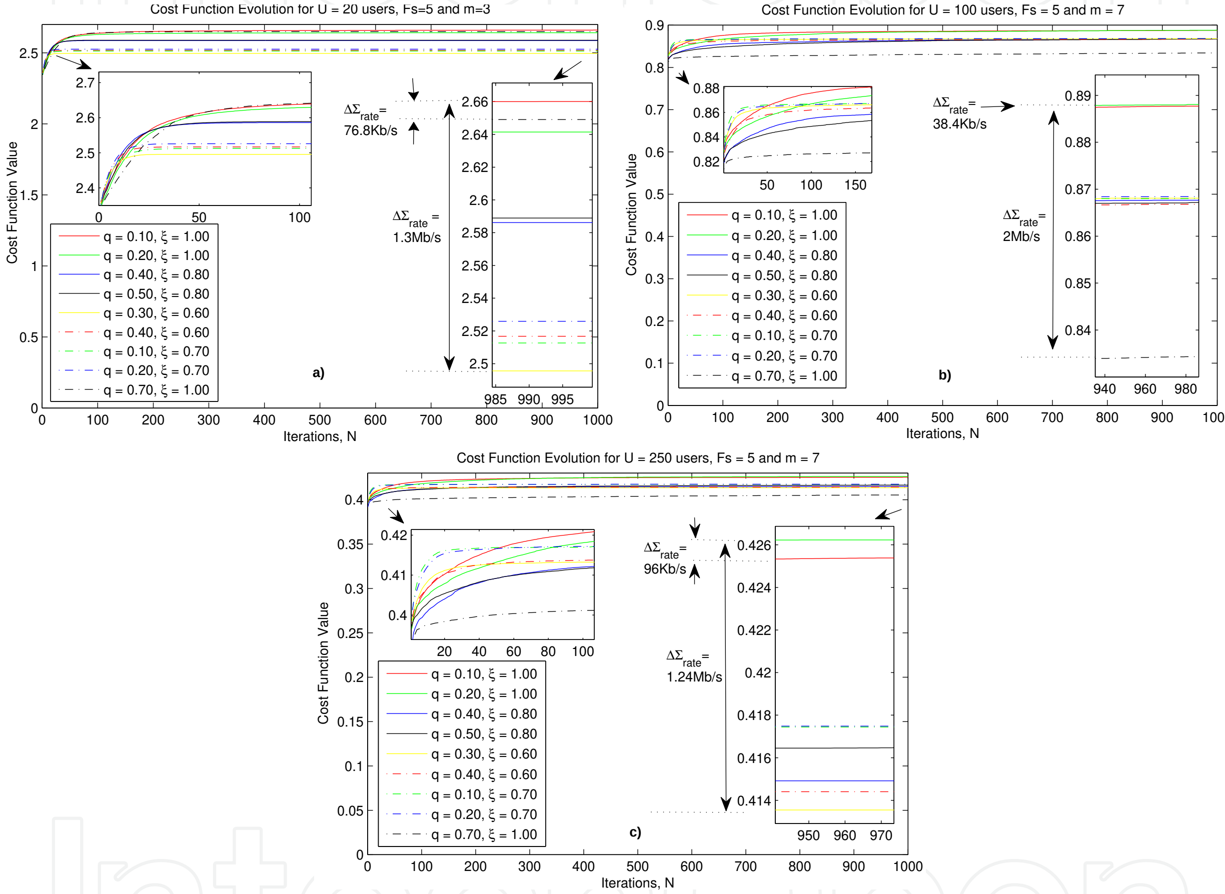


Figure 4. Cost function J evolution from Eq. (21) across $N = 1000$ iterations for different ACO input parameters values combination. The correspondent sum rate difference ($\Delta \sum_{\text{rate}}$) is zoomed in. a) $U = 20$; b) $U = 100$, and c) $U = 250$ users.

2.6.2. WTM RA-ACO Performance Results

Numerical results for the WTM problem with RA-ACO algorithm under optimized input parameters are shown in Figure 5. Here, its clear that ACO can evolve pretty fast to the three different system loads, finding a good solution in less than 100 iterations. Besides, one can note the great increase on the total system power from the 100 users case to the 250. It is due to the interference increase given the high number of users in the system. Nevertheless, a good system throughput result is found. For the 20 users results, a system total throughput of $200 Mb/s$ is found. This results in an remarkable average user rate of $10 Mb/s$.

On the $U = 100$ users results, $\approx 340 Mb/s$ of system throughput is reached, with a total power consumption of $\approx 55W$. Herein, the average user rate is $\approx 3,4 Mb/s$. This is due to the higher interference values given the medium system load.

Finally, for the $U = 250$ users results, a total system throughput of $\approx 400 Mb/s$ is reached, with a total power consumption of $\approx 125W$. Again, the average user rate decays regarding the low and medium system loads, reaching $\approx 1,6 Mb/s$.

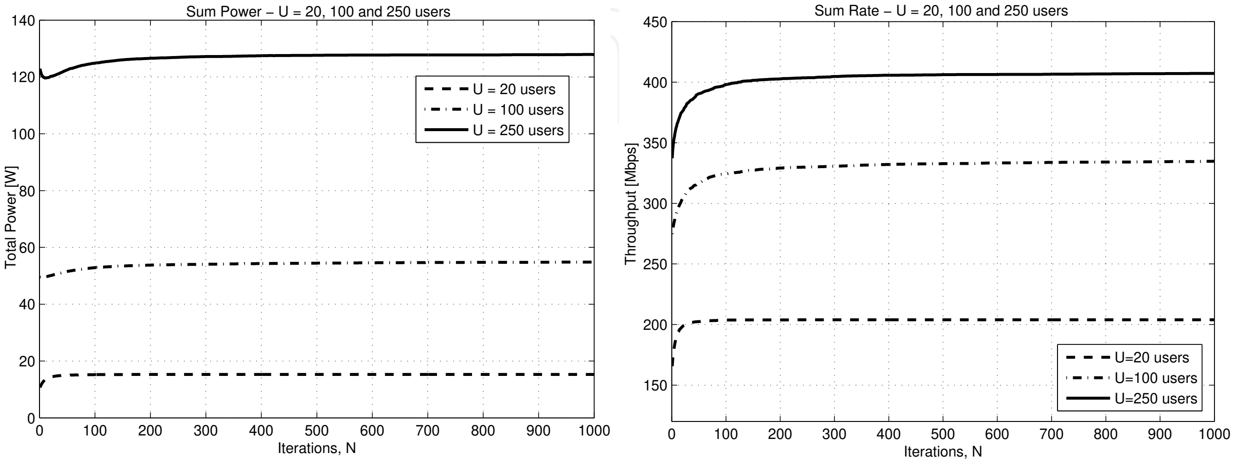


Figure 5. Sum Power and Sum Rate evolution for $U = 20, 100$ and 250 users under RA-ACO algorithm.

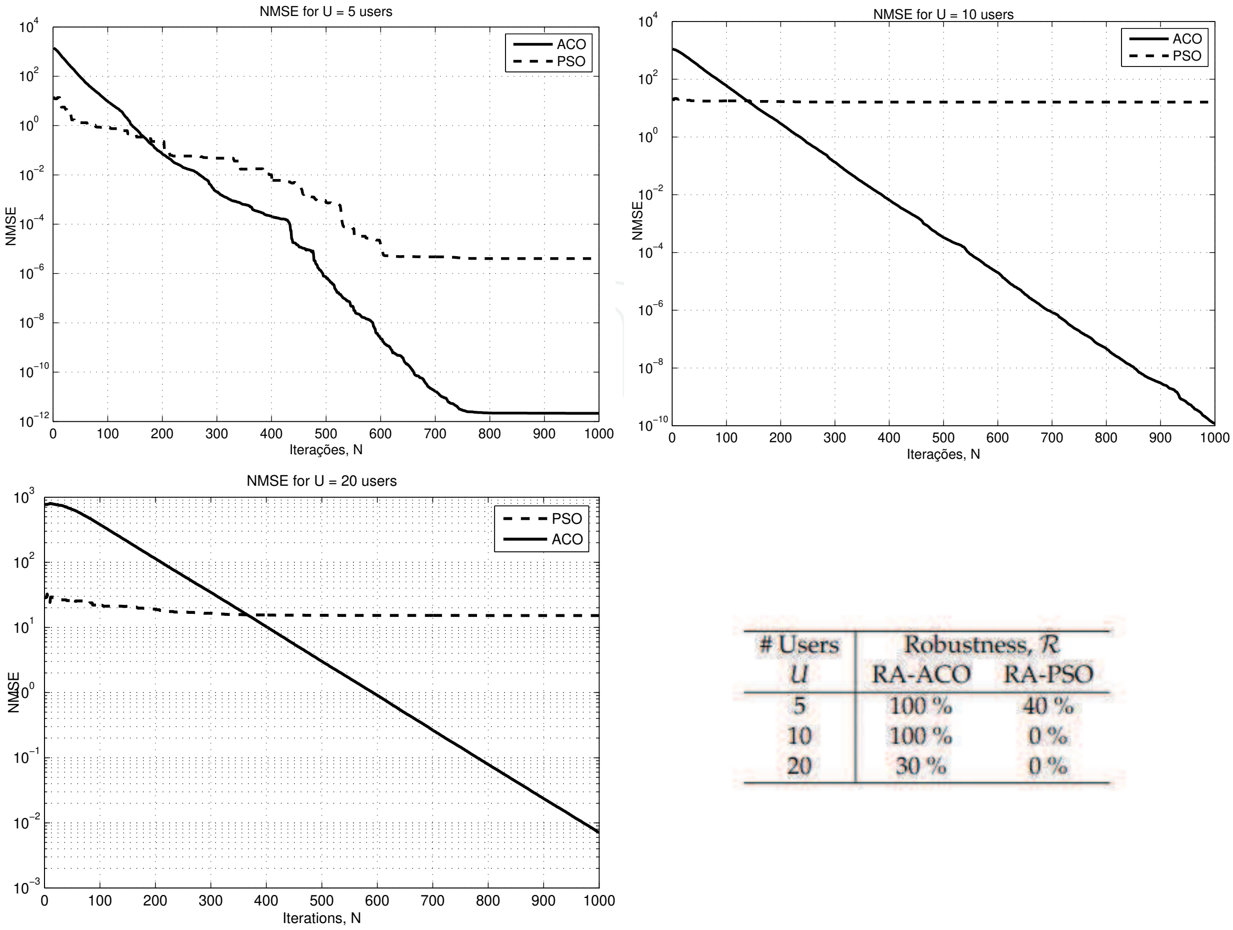
2.6.3. RA-ACO and RA-PSO Simulation Performance Results

The main objective in this analysis is put in perspective the both RA-heuristic algorithm performance regarding the non-convexity of the power allocation problem posed in (17). Simulations were conducted on different system loadings according to the best input RA-ACO parameters presented in Section 2.6.1 and those best parameters obtained in [1] for the resource (power) allocation using particle swarm optimization (RA-PSO) algorithm.

Figure 6, shows the NMSE evolution for the power control problem with $U = 5$, 10 and 20 users, respectively, for the algorithms RA-PSO [1] and RA-ACO. Clearly, the NMSE $\approx 10^{-12}$, 10^{-10} and 10^{-2} attainable by the RA-ACO is much more lower than that values reached by RA-PSO (NMSE $\approx 10^{-5}$, 10^{1} and 10^{1}) after $N = 1000$ iterations. This means the ACO could surpass the various convergence problems in solving the non-convex power control problem. The associated robustness shown that the RA-ACO achieves near to total convergence success, while the RA-PSO was not able to do. Fig. 6 also shows a table containing the percentage of algorithm success, i.e. the percentage of trials in which the algorithm ended with a NMSE less or equal to 10^{-2}, showing a clearly superiority of the RA-ACO scheme. Nonetheless, this robustness comes with a computational complexity increasing.

3. Anomaly Detection in Computer Networks

This section, presents the main concepts and related work in computer networks anomaly. It is presented anomalies type and their causes, the different techniques and methods applied into anomaly detection, as well as a compilation of recent proposals for detecting anomalies in networks using ACO and a case study.



# Users	Robustness, $\mathcal{R}$	
U	RA-ACO	RA-PSO
5	100 %	40 %
10	100 %	0 %
20	30 %	0 %

Figure 6. NMSE attainable by RA-ACO and RA-PSO [1] algorithms, for $U = 5, 10$, and 20 users.

Management is an essential activity in enterprizes, service providers and other elements for which the networks have become a necessity because the continued growth of networks have introduced an extensive options of services. The main goal of management it is to ensure the fewest possible flaws and vulnerabilities to not affect the operation of networks. There are several factors that can lead to an anomaly such as configuration errors, improper use by users and programming errors and malicious attacks among many other causes [20].

A tool to help the network management task are the Anomaly Detection System (ADS), which consist of a technique or a set of techniques to detect anomalies and report to the network administrator, helping to decide which action perform in each situation. Anomaly detection is not an easy task and brings together a range of techniques in several areas. The quality and safety of the service provided to end users are ongoing concerns and receive special attention in network traffic.

3.1. Anomaly

Thottan et. al [21] present two anomalies categories. The first one is related to network failures and performance problems, where the role of a malicious agent does not exist. The flash crowd is a typical example of this category, where a server receives a huge amount of not malicious client requests in the same period of time, congesting the server, i.e., a webstore promotes a product at a lower price at a certain time. If the webstore does not prepare a infrastructure to support the client access, the server may interrupt the operations and do not operate all sales transactions, causing congestion on the server and possible financial loss to the webstore. The congestion itself is another anomaly type within the first category, due to the abrupt increase in traffic at some point in the network, causing delays in the delivery of packages until the saturation of the link, with packet discards. Congestion can also be generated by configuration errors, the server does not respond adequately to requests sent by clients by wrong settings.

In the second category, it is found anomalies that arise from problems related to security. Denial of Service (DoS) is a main example of a anomaly in this category. DoS occurs when a user is unable to obtain a particular service because of some malicious agent used methods of attack that occupied the machine resources such as CPU, RAM. Besides DoS attacks, also have Distributed Denial of Service (DDoS) where a master machine dominates other machine, called zombies, to perform a DoS [22]. The flash crowd is differentiate from DoS and DDoS because of the malicious agent. Worms, port scan and others usually are programmed to discover vulnerabilities in networks and perform attacks [23].

3.2. Anomaly Detection Techniques

The techniques implemented in ADS are present in diverse areas such as intrusion detection, fraud detection, medical anomaly detection, prevention of damage to industrial image processing, sensor networks, among others [24]. Presenting so many different application domains, many tools have been developed specifically for some activity and other solutions are more general. Chandola et. al [24] group the techniques into: based on classification, clustering, information theory, statistical and spectral theory. [24, 25] are surveys with a general content in anomaly detection field, but it is found surveys toward the area of computer network, [20, 26]. The nomenclature and some categories may differ, but the concept presented is consistent with each other.

Patcha et. al [20] divides the techniques of detecting anomalies in three categories: based on signature, based on the characterization of normal behavior and hybrid techniques. The signature-based techniques are based on the number of signatures of known or constructed attack patterns. The strength of this kind of detection is the low rate of false positives. The techniques based on characterization of normal traffic build the profile of network traffic, and any event that deviates from the normal behavior is considered an anomaly. The hybrid techniques are a junction of the two previous techniques [20].

Many authors consider the proposed work by Denning [27] as a watershed between the methods based on signature and the methods used to characterize the normal traffic behavior, and these methods consist of two phases: training phase and test phase. The training phase is generated from the network profile and the test phase is applied the profile obtained to evaluation.

3.2.1. Detection Based on Signature

Detection based on signature requires the construction of a database of events related to certain anomalies, thereby generating the signature. The signatures describes specific events that form a specific attack or anomaly; this way, when the tool monitors the traffic behavior, the comparison with the signatures is performed, and if a match occurs as described the event in the signature, an alarm is generated [20].

Using signature, the method offers low rates of false positive, since the signature clearly describes what is required to be considered an anomaly, however, unknown attacks characteristics and not formulated signatures might pass unnoticed. Another negative point is the need to constantly update the signatures database [20].

3.2.2. Detection Based on the Characterization of Normal Behavior

Unlike the signature-based detection, the focus of this method is to detect anomalies based on the characterization of normal behavior. The first and fundamental step is to generate the normal behavior profile of traffic or adopt a model that more accurately describes the traffic. Consequently, any activity that deviates from the monitored normal profile built will be considered anomaly. The construction of the profile can be static or dynamic. It is static when the profile is built and replaced only when a new profile is constructed, and it is dynamic when the profile is updated according to the network behavior changes

A positive point is the possibility of detecting new anomalies, whereas these new anomalies describe a behavior different from normal. Another aspect is the difficulty created for the malicious agent devise an attack, because it ignores the profile and the possibility exists that it can not simulate an attack describing the profile and generates an alarm [20]. But there are disadvantages in profile construction as the required training period or the information amount on the basis of historical data. The difficulty in characterizing the traffic itself generates a high percentage rate of false positives, since the ADS can point to many natural variations of the network as an anomalous behavior.

There are several techniques, below are listed some relevant techniques that enrich the discussion with several different proposals:

- **Machine Learning:** The machine learning solutions have the ability to learn and improve the performance over time because the system changes the implementation strategy based on previous results. Bayesian networks, Markov chains, neural network are techniques applied to the generation of the normal profiles and detection of the anomalies. The main advantage of this approach is the ability to detect anomalies unknown and adapt to changes in the behavior of a monitored environment, however, this adjustment requires a large amount of data to generate a new profile [20].
- **Based on Specification:** These solutions are constructed by an expert, since the specifications of the normal behavior of the system are carried out manually. If the system is well represented, the false negative rate will be minimized by avoiding any behavior not predicted, but may increase if some behavior is overlooked or not well described. The most widely used technique for this task are finite state machines. A drawback of this approach point is the time and complexity to the development of the solutions [26].
- **Signal Processing:** The most commonly used techniques are the Fourier transforms, wavelet and algorithms such as ARIMA (Autoregressive Integrated Moving Average). It presents the advantage to adapt to the monitored environment and detecting unknown anomalies and low training period. The complexity is presented as a disadvantage of this approach [28].

- **Data Mining:** The Data Mining techniques usually deal with a huge amount of data, looking for patterns to form sets of normal data. Principal Component Analysis (PCA), clustering algorithms, Support Vector Machine (SVM) and others statistical tools are commonly employed in these solutions [20].

3.3. Recent Proposals Using ACO in Computer Networks Field

Since Dorigo et. al [29, 30] proposed the Ant System (AS) from the first time, several applications have emerged using AS itself or others algorithms arising from the ACO approach. One algorithm proposed to the networks routing problem is the AntNet, proposed by Di Caro et al. [31], a different approach to the adaptive learning of routing tables in communications networks. To the information to travel from point A to point B, it is necessary to determine the path that will be covered. The construction process itself and the path is known as routing, and it is one at the core of the To the network control system together with congestion control components, admission control, among others [31]. The AntNet is close to the real ants' behavior that inspired the ACO metaheuristic, because the routing problem can be characterized as a directed weighted graph, where the ants move on the graph, building the paths and loosing the pheromone trails.

Information Retrieval is another field where ACO found application, as proposed by [32, 33]. The problem in the information retrieval system consists in finding a set of documents including information expressed in a query specifying user needs. The process involves a matching mechanism between the query and the documents of the collection. In [32], Drias et. al designed two ACO algorithms, named AC-IR and ACS-IR. Each term of the document has an amount of pheromone that represents the importance of its previous contribution in constructing good solutions, the main difference between AC-IR and ACS-IR is mainly in the design of the probabilistic decision rule and the procedure of building solutions. In [33], the ACO algorithm is applied to retrieve relevant documents in the reduced lower-dimensionality document feature space, the probability function is built using the frequency of the terms and the total number of documents containing the term.

In [34], the autors make use of a Fuzzy Rule Based System (FRBS), Naive Bayes Classifier (NBC) and Support vector machine (SVM) to increase the interpretability and accuracy of intrusion detection model for better classification results. The FRBS system is a set of IF-THEN rules, whose antecedents and consequents are composed of fuzzy statements, related by the dual concepts of fuzzy implication and the compositional rule of inference. The NBC method based on the "Bayes rule" for conditional probability as this rule provides a framework for data assimilation. The SVM is a statistical tool for data classification which is one of the most robust and accurate methods among all well-known algorithms. Its basic idea is to map data into a high dimensional space and find a separating hyper plane with the maximal margin. Then, the authors proposed NBC with ACO, linking a Quality computation function, ranking the best rule between discovered ones, to the pheromone updating.

A commonly approach to network intrusion detection is to produce cluster using a swarm intelligence-based clustering. Therefore, in the traditional clustering algorithms it is used a simple distance-based metric and detection based on the centers of clusters, which generally degrade detection accuracy and efficiency because the centers might not be well calculated or the data do not associate to the closest center. Using ACO, it is possible to surround the local optimum and find the best or the most close to the best center. This technique is used in [35], Feng et. al present a network intrusion detection, assuming two assumptions: the number of normal instances vastly outnumbers the number of intrusions and the intrusions themselves are qualitatively different from the normal instances. Then, three steps are followed: 1) Clustering, 2) Labeling cluster and 3) Detection.

In [36], it is found the use of data mining to intrusion detection. Abadeh et. al proposed an extract fuzzy classification rules for misuse intrusion detection in computer networks, named Evolutionary Fuzzy System with an Ant Colony Optimization (EFS-ACO). It consists of two stages, in the first stage, an iterative rule learning algorithm is applied to the training data to generate a primary set of fuzzy rules. The second stage of the algorithm employs an ant colony optimization procedure to enhance the quality of the primary fuzzy rule set from the previous stage.

3.4. Applying ACO for Anomaly Detection - A Study Case

This sections provides an application of ACO for anomaly detection. The proposed approach is under the categorie of the detection methods based on the characterization of normal behavior, and follow two steps: 1) Training Phase, 2)Detection Phase. The dataset used for evaluation of the method is the KDD'99 [37]. In the Training Phase, it is used the training dataset and it is generated the centroids for each class of attack, and in the Detection Phase, it is used the generated centroids in each conection to classify it and generate the final resuts.

3.4.1. KDD Cup 99 Data Set Description

For evaluation for anomaly detection methods it is commonly used the KDD'99 dataset, used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining [37]. This dataset was built based on the data captured in DARPA'98 IDS evaluation program, used for The Second International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-98 The Fourth International Conference on Knowledge Discovery and Data Mining [38].

KDD training dataset consists of approximately 490,000 single connection vectors each of which contains 41 features and it is labeled as: back, buffer overflow, ftp write, guess passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster, normal. Depending on the label, the connection fall in one of the following four attack categories:

1. **Denial of Service (DoS)**: is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
2. **User to Root (U2R)**: is a class of exploit in which the attacker starts out with access to a normal user account on the system and is able to exploit some vulnerability to gain root access to the system.
3. **Remote to Local (R2L)**: occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

4. **Probing** : is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

In Table 4 it is present the labels related to the attack categories. From all the attack categories, the study subject will be the DoS attacks. From all the 41 features, for this study case, it is adopt the source bytes and destiny bytes, because the main idea of the approach is to detect volume anomaly.

Attack Categorie	Labels	Samples
Denial of Service (DoS)	back, land, neptune, pod, smurf, teardrop	391458 (79.2391%)
User to Root (U2R)	buffer overflow,perl, loadmodule, rootkit	52 (0.0105%)
Remote to Local (R2L)	ftp write, guess passwd, imap, multihop, phf, spy, warezclient, warezmaster	1126 (0.2279%)
Probing	ipsweep, nmap, portsweep, satan	4107 (0.8313%)

Table 4. Labels related to the attack categories

3.4.2. The ACO Clustering

ACO is composed of a population of agents competing and globally asynchronous, cooperating to find an optimal solution. Although each agent has the ability to build a viable solution, as well as a real ant can somehow find a path between the colony and food source, the highest quality solutions are obtained through cooperation between individuals of the whole colony. Like other metaheuristics, ACO is compound of a set of strategies that guide the search for the solution. It makes use of choices based on the use of stochastic processes, verifying the information acquired from previous results to guide it through the search space [39].

Artificial ants travel the search space represented by a graph $G(V, E)$, where V is a finite set of all nodes and E is the set of edges. The ants are attracted to more favorable locations to optimize an objective function, in other words, those in which the concentration of pheromone deposited by ants that previously went through the same path is higher [40]. While real ants deposit pheromone on the place they visit, artificial ants change some numeric information stored locally which describe the state of problem. This information is acquired through the historical and current performance of the ant during construction of solutions [39].

The responsibility of hosting the information during the search of the solution lies with the trail pheromone, τ. In ACO, the tracks are channels of communication between agents and only they have access to the tracks, i.e., only ants have the propriety of reading and modifying the numeric information contained in the pheromone trails. Every new path selection produces a solution, and each ant modifies all local information in a given region of the graph. Normally, an evaporation mechanism modifies the pheromone trail information over time. This characteristic allows agents slowly forget their history, allowing their search for new directions without being constrained by past decisions, thereby, avoiding the problem of precipitated convergences and resulting in not so great solutions.

The technique of clustering is a data mining tool used to find and quantify similarities between points of determined group of data. This process seeks to minimize the variance between elements of a given group and maximize them in relation to other groups [41]. The similarity function adopted is the Euclidean distance described in Eq. (28).

$$d(x,y) = \sqrt{\sum_{i=1}^{m} |x_i - y_i|^2} = \|\mathbf{x}_i - \mathbf{y}_i\| \tag{28}$$

The equation that measures the similarity between the data is called the objective function. The purpose of the use clustering is to create a template from which to extract a pattern of information. Thus, when a distance of data is found in smaller quantities in relation to this standard, you can group them into clusters of different sets of greater representation. The most classical algorithm in the literature is the K-means (KM) algorithm. It is a partitional center-based clustering method and the popularity is due to simplicity of implementation and competence to handle large volumes of data.

The problem to find the K center locations can be defined as an optimization problem to minimize the sum of the Euclidean distances between data points and their closest centers, described in Eq. (29). The KM randomly select k points and make them the initial centres of k clusters, then assigns each data point to the cluster with centre closest to it. In the second step, the centres are recomputed, and the data points are redistributed according to the new centres. The algorithm stop when the number of iterations is achieved or there is no change in the membership of the clusters over successive iterations [42]. One issue founded in KM is the initialization due to partitioning strategy, when in local density data results in a strong association between data points and centers [43].

$$\mathrm{KM}(\mathbf{x},\mathbf{c}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \|x_i - c_j\|^2 \tag{29}$$

where $\mathbf{x}$ is the data, $\mathbf{c}$ is the center. The parameter n is total number of elements in $\mathbf{x}$ and k is the number of center in $\mathbf{c}$.

The ACO described in this section aims to optimize the efficiency of clustering minimizing the objective function value described by Eq. (29). Thus, this ensures that each point i will be grouped to the best cluster j where $j = 1, ..., K$. In addition, it enables the construction of solutions that are not givens by local optimal, which is the existing problem in most clustering algorithms. The ACO algorithm proposed is described in Algorithm 2, in which the functions performed into the Algorithm 2 are briefly described in following.

a) *CalculateFitnessFunction()*: For each ant m is calculated the Fitness Function based on Eq. 29. As each ant represent a possible solution, each ant will be a possible center to clusterize the data $\mathbf{x}$, and the ant m describing the lowest value of $\mathrm{KM}(\mathbf{x}, \mathbf{c}_m)$.

b) *SortAnts()*: This function sort and rank the ants according to the *CalculateFitnessFunction()*.

Algorithm 2 ACO Clustering

```
Objective function $f(\mathbf{x}), \mathbf{x} = (x_m, ..., x_d)^T$
Initialize the ants population $\mathbf{x}_m(m = 1, 2, ..., n)$
Set the parameters $\gamma, \beta, \rho$
WHILE (The end conditions aren't met)
  FOR $m = 1$ to $M$
      CalculateFitnessFunction();
  end FOR
  SortAnts();
  UpdatePheromone();
end WHILE
```

c) *UpdatePheromone()*: This function directs the algorithm at the search for new solutions using promising path that were previously found. The links between point-cluster that showed better results are intensified and expected to be used in the construction of increasingly better solutions. In contrast, point-cluster unsuccessful links are expected to be forgotten by the algorithm through the evaporation process of the pheromone. The pheromone updating can be described by:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) \tag{30}$$

where ρ is a constant suitably defined, which describes the evaporation rate of the pheromone and has value $0 < \rho < 1$. The variable t identifies the interaction running.

After the ACOClustering generates the centers from the training dataset, it is applied to the dataset. Then, it is adopted a parameter ϵ which is a value describing a range accepted to cluster the data in that center of not. The figure 7 illustrated the idea.

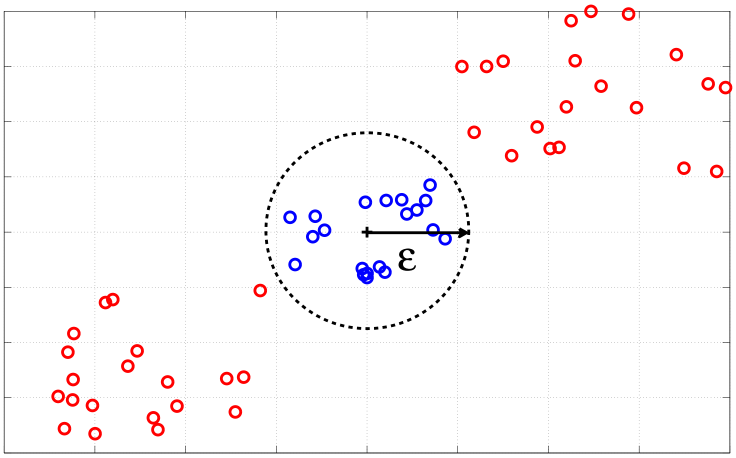


Figure 7. The area generated by the ϵ parameter.

3.4.3. Numerical Results

A paramount importance question when working with cluster is the optimal number of clusters to grouping the dataset in a good manner. We adopted the following clustering quality criteria: Dunn's index [44] and Davies-Bouldin index [45].

The Dunn's index is based on the calculation of the ratio between the minimal intracluster distance to maximal intercluster distance and the main idea is to identifying the cluster sets that are compact and well separated. The following Eq. (31) describes:

$$D = \frac{d_{\text{min}}}{d_{\text{max}}} \tag{31}$$

where, d_{min} is the smallest distance between two objects from different clusters, and d_{max} is the largest distance of two objects from the same cluster. D is limited to the interval $[0, \infty]$ and higher is the desirable value. In figure 8, it is presented the values for the tests for $K = [2, \ldots, 10]$.

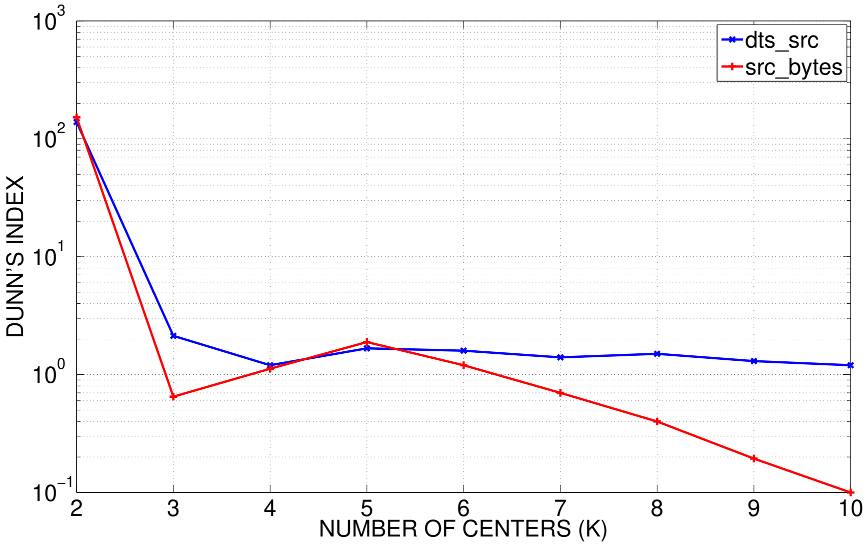


Figure 8. Dunn's index.

The Dunn's index aims to indentify clusters that are compact, well separeted and with a low variance between the members within the same cluster. The results indicates that $K = 2$ are the best number of centers to adopt, because it is the higher Dunn's index indicating a better clustering. But, before adopting $K = 2$, we tested another index, the Davies-Bouldin index [45]. It is a function of the ratio of the sum of within-cluster scatter to between-cluster separation and is describe by the Eq. (32):

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} \max \left[\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right] \tag{32}$$

where n is the number of clusters, σ_i is the average distance of all objects in cluster i to their cluster center c_i, σ_j is the average distance of all objects in cluster j to their cluster center c_j, and $d(c_i, c_j)$ is the distance of centers c_i and c_j. If DB result in low values, the clusters are compact and the centers are far from each other. In figure 9, it is presented the results for $K = [2, \ldots, 10]$.

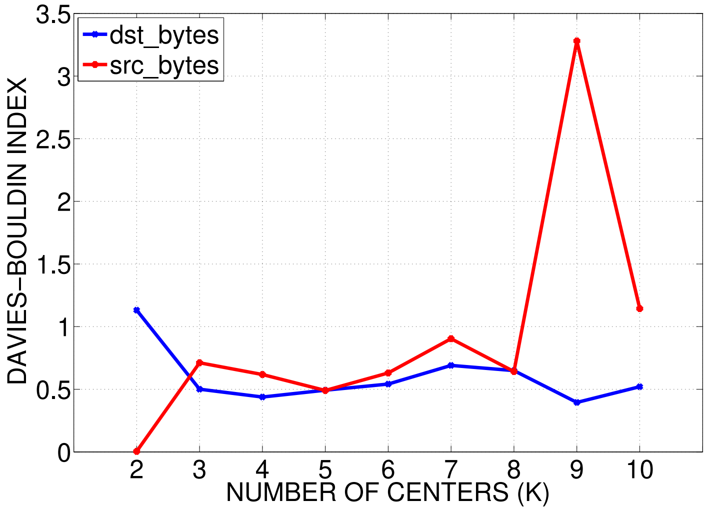


Figure 9. Davies-Bouldin index.

The ratio of the within cluster scatter to the between cluster separation will constrain the index to be symmetric and non-negative, thus it is expected a lower value for a fair clustering. From figure 9, the src_bytes (red line) present a value close to 0 at $K = 2$, and to the dst_bytes (blue line) for all the K tested values it had a regular behavior around 0.5. As $K = 2$ present a better result for Dunn's index and the best result for src_bytes, it is adopted for all the following tests.

Besides the number of centers, to measure the efficiency of the proposed case study, we adopted the following variables [46]:

- TRUE POSITIVE : If the instance is an anomaly and it is classified as an anomaly;
- FALSE NEGATIVE : If the instance is an anomaly and it is classified as normal;
- FALSE POSITIVE : If the instance is normal and it is classified as an anomaly;
- TRUE NEGATIVE : If the instance is an normal and it is classified as normal;

Hence, through the declaration of these variables the following equations can be calculated:

$$\text{False Alarm Rate (FAR)} = \frac{\text{FALSE NEGATIVE}}{\text{TOTAL OF NORMAL DATA}} \tag{33}$$

$$\text{Accuracy (ACC)} = \frac{\text{TRUE POSITIVE} + \text{TRUE NEGATIVE}}{\text{TOTAL NORMAL DATA + TOTAL ANOMALY DATA}} \tag{34}$$

$$\text{Precision (PRE)} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE POSITIVE}} \tag{35}$$

Eq. (33) describes how much the method wrongly classified as anomalous from all the normal data, while Eq. (34) measures the closeness of the method measures in relation to the real values. The Eq. (35) describes the percentage of the corrected classified data among all the classified data.

It was decided to test four rules to capture the anomalies: 1)using only the src_bytes, 2)using only the dst_bytes, 3)using the src_bytes or dst_bytes and 4)using the src_bytes and dst_bytes. The figure 10 shows the results for the FAR.

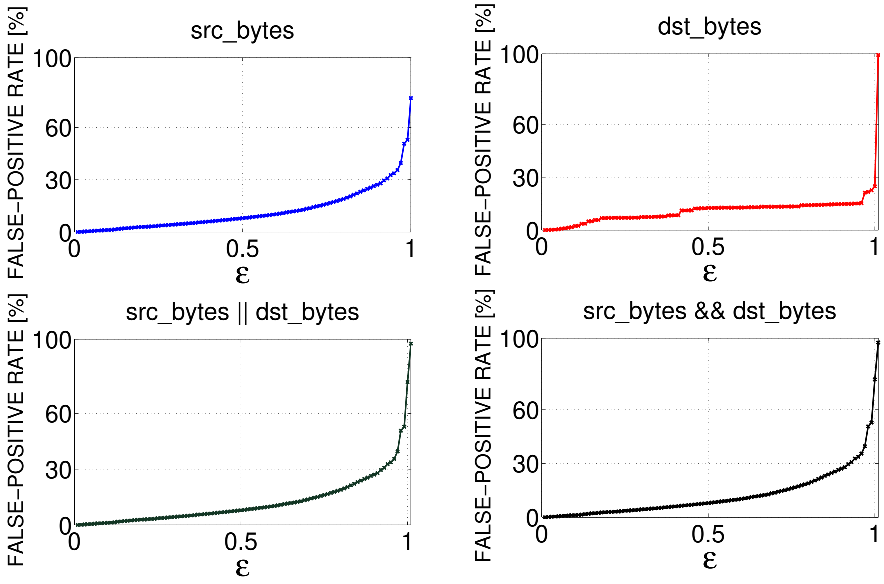


Figure 10. False alarm rate.

The figure 10 shows the method study achievies low rates for FAR, that means the method does not classify as anomalous the normal data. As we increase the value of ϵ, the FAR starts increasing, but only achieves high rates close to 1, therefore, the area created is large enough to capture anomalies data and wrongly classify.

In figure 11 shows the result for ACC. The method is not so close to the real value, expressed by the rate around 30%. This rate can be originated because the method is not classifying right the anomalies, in the other hand, it is classifying the normal data right.

To finally demonstrate that the method is not classifying the anomalies in a good manner, the figure 12 shows the precision results. It is observed higher rates when $\epsilon < 0.15$, meaning the method can classify anomaly from normal when the area adopted is small. This makes sense, because in computer networks the traffic behavior follows a regular action and the anomaly usually is a abrupt change in this behavior. When $\epsilon > 0.15$, the PRE rate decrease at a high pace, that means the method adopt more anomalies as normal data.

As for the four rules, we can conclude that when separate they show different results, the src_bytes describes better results. But when using then together, they express similar results as well, and the src_bytes results suppress the dst_bytes results.

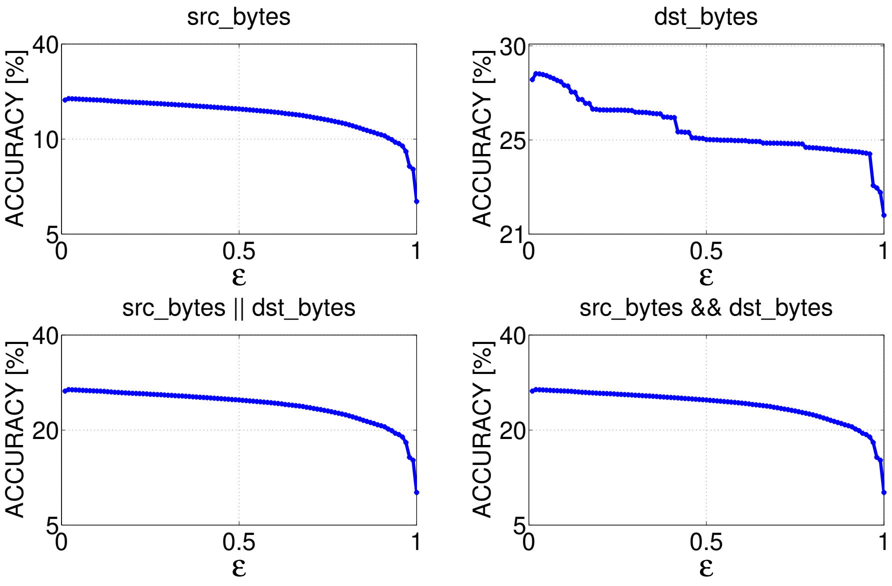


Figure 11. Accuracy rate.

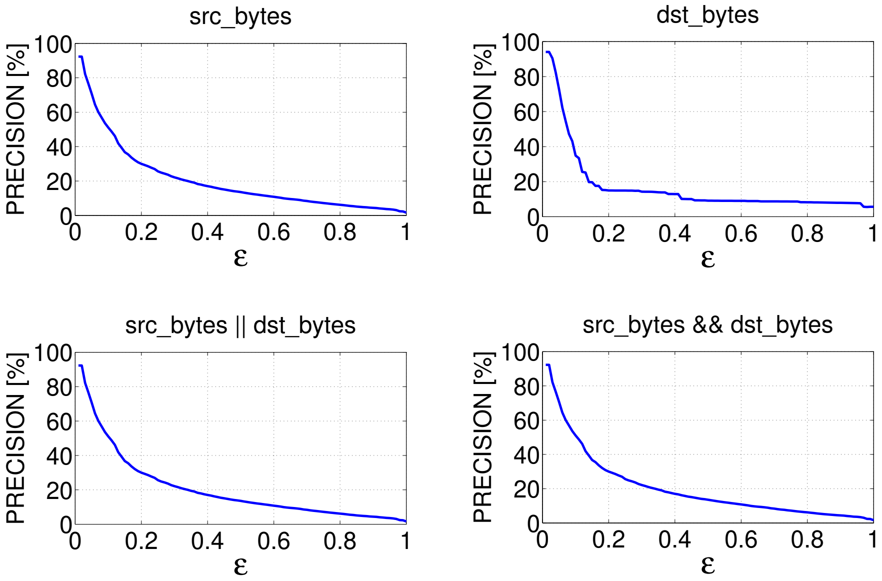


Figure 12. Precision rate.

4. Conclusion Remarks

4.1. ACO Resource Allocation in CDMA Networks

The ACO algorithm proved itself robust and efficient in solving both RA problems presented in this chapter. In fact, the ACO performance exceeded the PSO performance discussed in [1].

In terms of solution quality the ACO power control scheme was able to achieve much better solutions than the PSO approach. For the weighted throughput maximization problem, the numerical results show a fast convergence in the first hundred iterations and a solution improvement after that. This fast convergence behavior is an important feature due to the system dynamics, i.e. the algorithm must update the transmission power every 666.7μs if we consider power control applications in 3G wireless cellular communication CDMA systems.

Future work includes analyzes over dynamic channels, i.e. the channel gain matrix is constant only over a single time slot.

4.2. ACO Anomaly Detection in Computer Networks

The method presented in the study case does not have excellent results, because for all the 41 features from the KDD dataset, it only works with 2: src_bytes and dst_bytes. From all the anomalies presented, we focus on the Denial of Service (DoS), and the method presented good False Alarm Rates (FAR), assuming the parameter ϵ normally is $0 < \epsilon < 2$, where the FAR is below 5%.

The Accuracy rate and Precision rate can be increased using the other features like flag, land. Flag is the status the connection and land assumes 1 if connection is from/to the same host/port; 0 otherwise. Adding more rules to the anomaly detection method, it is possible to increase the rates.

The ACOClustering have an important rule, helping to cluster the data, preventing to get stuck in local optimum centers. In the task of manage the computer network, it can handle a set of thousands of connections per second, thus it is possible to get stuck in some optimum local center. As each ant is a possible set of the centers, and the search is always guide by the best answer found so far.

Author details

Mateus de Paula Marques[1],
Mário H. A. C. Adaniya[1], Taufik Abrão[1],
Lucas Hiera Dias Sampaio[2] and Paul Jean E. Jeszensky[2]

1 State University of Londrina (UEL), Londrina, PR, Brazil
2 Polytechnic School of University of São Paulo (EPUSP), São Paulo, SP, Brazil

References

[1] T. Abrão, Sampaio L.D.H., M. L. Proença JR, B. A. Angélico, and P. J. E. Jeszensky. *Multiple Access Network Optimization Aspects via Swarm Search Algorithms*, volume 1, chapter 13, pages 261–298. InTech Open, 2011.

[2] Marco Dorigo and Gianni Caro. Ant colony optimization: A new meta-heuristic. In *Evolutionary Computation. CEC 99.* IEEE, 1999.

[3] Gerhard Fettweis and Ernesto Zimmermann. Ict energy consumption - trends and challenges. In *WPMC'08 – 11th International Symposium on Wireless Personal Multimedia Communications*, Sept. 2008.

[4] G. Foschini and Z. Miljanic. A simple distributed autonomous power control algorithm and is convergence. volume 42, pages 641–656. IEEE, November 1993.

[5] M. Moustafa, I. Habib, and M. Naghshineh. Genetic algorithm for mobiles equilibrium. MILCOM 200, October 2000.

[6] M.Moustafa, I. Habib, and M. Naghshineh. Wireless resource management using genethic algorithm for mobiles equilibrium. volume 37, pages 631–643, November 2011.

[7] H. Elkamchouchi, H. Elragal, and M. Makar. Power control in cdma system using particle swarm optimization. pages 1–8, March 2007.

[8] K. Zielinski, P. Weitkemper, R. Laur, and K. D. Kammeyer. Optimization of power allocation for interference cancellation with particle swarm optimization. volume 13, pages 128–150, February 2009.

[9] J.W. Lee, R.R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class wireless systems. volume 13, pages 854–867. IEEE, August 2005.

[10] J. Dai, Z. Ye, and X. Xu. Mapel: Achieving global optimality for a non-convex wireless power control problem. volume 8, pages 1553–1563. IEEE, March 2009.

[11] T. J. Gross, T. Abrao, and P. J. E. Jeszensky. Algoritmo de controle de potência distribuido fundamentado no modelo populacional de verhulst. volume 20, pages 59–74. Revista da Sociedade Brasileira de Telecomunicacoes, 2010.

[12] J. H. Ping Qian and Ying Jun Zhang. Mapel: Achieving global optimality for a non-convex wireless power control problem. volume 8, pages 1553–1563, March 2009.

[13] Lucas Dias H. Sampaio, Moisés F. Lima, Bruno B. Zarpelão, Mario Lemes Proença Junior, and Taufik Abrão. Swarm power-rate optimization in multi-class services ds/cdma networks. 28th Brazilian Symposium on Computer Networks and Distributed Systems, May 2010.

[14] M. Elmusrati and H. Koivo. Multi-objective totally distributed power and rate control for wireless communications. volume 4, pages 2216–2220. VTC'03-Spring, Apr. 2003.

[15] C. E. Shannon. The mathematical theory of communication. *The Bell System Technical Journal*, 27((reprinted with corrections 1998)):379–423, 623–656, July, October 1948.

[16] M. Elmusrati, H. El-Sallabi, and H. Koivo. Aplications of multi-objective optimization techniques in radio resource scheduling of cellular communication systems. volume 7, pages 343–353. IEEE, Jan 2008.

[17] E. Seneta. *Non-Negative Matrices and Markov Chains*, volume 2. Springer-Verlag, 1981.

[18] N. T. H. Phuong and H. Tuy. A unified monotonic approach to generalized linear fractional programming. pages 229–259, 2003.

[19] Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. In *European Jornal of Operational Research*, pages 1155–1173, Brussels, Belgium, 2008. Elsevier.

[20] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51:3448–3470, August 2007.

[21] M Thottan and Chuanyi Ji. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, August 2003.

[22] Wentao Liu. Research on DoS attack and detection programming. In *Proceedings of the 3rd international conference on Intelligent information technology application*, volume 1 of *IITA'09*, pages 207–210, Piscataway, NJ, USA, November 2009. IEEE Press.

[23] J. Gadge and A.A. Patil. Port scan detection. In *16th IEEE International Conference on Networks*, ICON, pages 1–6, USA, December 2008. IEEE Press.

[24] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys.*, 41(3), 2009.

[25] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.

[26] Juan M. Estévez-Tapiador, Pedro Garcia-Teodoro, and Jesús E. Díaz-Verdejo. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, 27(16):1569–1584, 2004.

[27] D.E. Denning. An intrusion-detection model. *Software Engineering, IEEE Transactions on*, SE-13(2):222–232, February 1987.

[28] Bruno Bogaz Zarpelão. *Detecção de Anomalias em Redes de Computadores*. PhD thesis, Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Eletrica e de Computação (FEEC)., 2010.

[29] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Positive feedback as a search strategy. Technical report, Technical Report No. 91-016, Politecnico di Milano, Italy, 1991.

[30] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1):29–41, 1996.

[31] Gianni Di Caro and Marco Dorigo. Antnet: Distributed stigmergetic control for communications networks. *J. Artif. Intell. Res. (JAIR)*, 9:317–365, 1998.

[32] Habiba Drias, Moufida Rahmani, and Manel Khodja. Aco approaches for large scale information retrieval. In *World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 713–718. IEEE, December 2009.

[33] Wang Ziqiang and Sun Xia. Web document retrieval using manifold learning and aco algorithm. In *Broadband Network Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on*, pages 152–155, oct. 2009.

[34] Namita Shrivastava and Vineet Richariya. Ant colony optimization with classification algorithms used for intrusion detection. In *International Journal of Computational Engineering and Management, IJCEM*, volume 7, pages 54–63, January 2012.

[35] Yong Feng, Zhong-Fu Wu, Kai-Gui Wu, Zhong-Yang Xiong, and Ying Zhou. An unsupervised anomaly intrusion detection algorithm based on swarm intelligence. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 3965–3969, aug. 2005.

[36] Mohammad Saniee Abadeh, Hamid Mohamadi, and Jafar Habibi. Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. *Expert Syst. Appl.*, 38(6):7067–7075, 2011.

[37] The UCI KDD Archive. Kdd cup 1999 data, 1999.

[38] The UCI KDD Archive. Kdd cup 1998 data, 1998.

[39] Marco Dorigo and Thomas Stützle. *Ant colony optimization*. MIT Press, 2004.

[40] P.S Shelokar, V.K Jayaraman, and B.D Kulkarni. An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2):187–195, 2004.

[41] Hui Fu. A novel clustering algorithm with ant colony optimization. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 66–69, dec. 2008.

[42] D. T. Pham, S. Otri, A. A. Afify, M. Mahmuddin, and H. Al-Jabbouli. Data clustering using the bees algorithm. In *Proc. 40th CIRP Int. Manufacturing Systems Seminar*, Liverpool, 2007.

[43] Fengqin Yang, Tieli Sun, and Changhai Zhang. An efficient hybrid data clustering method based on k-harmonic means and particle swarm optimization. *Expert Syst. Appl.*, 36(6):9847–9852, 2009.

[44] J.C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[45] David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.

[46] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2005.