Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

A Statistical Analysis of Self-Chord: on Possible Connections between Ant Algorithms and Random Matrix Theory

C. Mastroianni

RT-ICAR-CS-10-02 **May 2010**

Consiglio Nazionale delle Ricerche – Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sedi di Cosenza (*Via P. Bucci 41C, 87036 Rende, Italy*), di Napoli (*Via P. Castellino 111, 80131 Napoli*) e Palermo (*Viale delle Scienze, Parco D'Orleans, 90128 Palermo*)– URL: *www.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

A Statistical Analysis of Self-Chord: on Possible Connections between Ant Algorithms and Random Matrix Theory

C. Mastroianni

Rapporto Tecnico N.:
RT-ICAR-CS-10-02

Data:
May 2010

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

A Statistical Analysis of Self-Chord: on Possible Connections between Ant Algorithms and Random Matrix Theory

Carlo Mastroianni

ICAR-CNR,
Via P. Bucci 41C, 87036 Rende(CS), Italy
mastroianni@icar.cnr.it

Abstract. This paper reports on the results of a statistical analysis of the behavior of Self-Chord, a self-organizing P2P system in which the resource keys are dynamically sorted with an ant algorithm, without requiring centralized management or human intervention. Key sorting is driven by the values of "peer centroids", which summarize the values of the keys stored by respective peers. It emerged that the distribution of the spacings between peer centroids is very similar to the distribution of the eigenvalues of random matrices taken from the Gaussian Unitary Ensemble (GUE) in the limit of large matrix size. This empirical observation is also supported by several qualitative considerations on the similarity between the behavior of Self-Chord centroids and that of the energy levels of physical systems modeled by GUE matrices. Since Self-Chord uses a basic and canonical ant algorithm, this analysis seems to suggest the hypothesis that the mathematical nature of ant algorithms is inherently connected to random matrix theory and, more widely, to number theory.

Key words: Ant algorithm, GUE operators, Random matrices, P2P

1 Introduction

Peer-to-peer (P2P) techniques and algorithms are steadily emerging as efficient solutions for the management of large-scale distributed computing systems, such as Grid and Cloud frameworks. In particular, the P2P paradigm is used to cope with the placement, advertising and discovery of resources needed by users for the execution of their applications.

In recent years, there have been very interesting attempts to reinforce the adaptive and fault-tolerance characteristics of P2P networks by imitating the self-organizing behavior of biological systems, such as flocks of birds, insect swarms, and, above all, ant colonies [9][10]. These algorithms exploit the properties of "swarm intelligence" systems, in which an intelligent behavior at a high level is obtained by combining simple low level operations performed by bio-inspired mobile agents [1][17]. The use of self-organizing algorithms has triggered the design of novel P2P systems that combine the benefits of structured and unstructured approaches[1]. These P2P systems are sometimes referred to as "self-structured" [7][2], because the organization and ordering of resources does not rely upon a predetermined network structure, but is obtained through the use of self-organizing techniques.

An example of this kind of systems is Self-Chord [8][6]. Self-Chord exploits the ring-shaped overlay of peers offered by the renowned Chord system [16], but it breaks the tight correlation between resource keys and peer indexes: the keys are not assigned to specified peers, as in Chord, but are sorted over the ring, in a self-organizing fashion, through "pick" and "drop" operations performed by ant-inspired agents. These operations are performed as the result of Bernoulli trials whose probabilities depend on the distance between the keys under considerations and the "centroid" of the local peer. The value of the peer centroid is used to summarize the key values stored in the peer: agent operations tend to pick keys that are distant from the local centroid and move them to the region of the ring where the peers have centroids similar to the key. With these simple operations, both the centroids and the keys are sorted over the ring.

The sorting of keys allows discovery operations to be executed in logarithmic time, thus preserving the fundamental service offered by Chord. In addition, Self-Chord offers further notable advantages, among which a better load balance and an improved capacity for recovering rapidly from perturbations. More importantly, since the space of resource keys and the space of key indexes are independent from each other, in Self-Chord keys can be given a semantic meaning, and it is possible to place similar keys onto neighbor nodes. This greatly helps the management of range queries, very important in distributed systems.

Even in a steady situation, the ants operate continuously to cope with possible perturbations, caused by network churns (peers that leave/join the ring) or by the publication/removal of keys. Therefore, the values of peer centroids are always modified by environmental changes, and yet the statistical distribution of peer centroids is very stable. This distribution is worth being analyzed, because the efficiency of discovery operations depends on the centroids being sorted and equally spaced over the ring. The statistical analysis led to a very surprising result. The distribution of the spacings between the centroids of consecutive peers appeared to be very similar to the distribution of the spacings between the eigenvalues of an important set of random matrices: the Gaussian Unitary Ensemble (GUE). The operators associated with this kind of matrices are used to model a wide class of complex dynamical systems, especially in the domain of nuclear physics [13], and these systems are characterized by a semi-chaotic response to external perturbations.

This paper reports on the results of a statistical analysis that confirmed the mentioned similarity. Besides confirming the robustness of the key sorting in Self-Chord, this study aims to be a stimulus for researchers in the field of ant algorithms, by asking questions such as: is this similarity peculiar to this

[1] In structured P2P systems each resource is consigned to a peer whose position in a predefined structure is obtained with a hash function. In unstructured systems, resources are positioned and managed without reference to any given structure.

particular case, related to P2P, or does it apply for a wide class of ant algorithms? then, can ant algorithms be analyzed by means of random matrix theory? is it possible to define a GUE operator that models the behavior of ants and helps to predict the effect of ant operations?

These questions, and their possible answers, are discussed at the end of the paper, in Section 5. Beforehand, Section 2 summarizes the key points of the Self-Chord P2P system, Section 3 gives a brief overview into the theory of random matrices, and Section 4 analyzes the statistical analogies between the distribution of Self-Chord centroids and that of GUE eigenvalues.

2 Self-Chord: an Ant-Inspired P2P System

In Self-Chord, peers are organized in a logical ring. Each peer is given an index of B_p bits, obtained with a uniform hash function. The ring is constructed and maintained as in Chord, see [16] for the details. Each resource is associated with a key, having N_c possible values, which will be used to discover and access the resource. Contrary to Chord, the value of N_c can be different from the number of admissible peer indexes, $N_r = 2^{B_p}$. This allows resource keys to be decoupled from peer indexes, with all the related advantages, in particular the possibility of assigning a semantic meaning to resource keys.

For their work, the agents use the concept of peer *centroid*. The centroid of a peer is defined as the real value, between 0 and N_c, which minimizes the average distance between itself and the keys stored by the peer. For example, with N_c=64, a peer that stores three keys with values $\{4,6,8\}$ has a centroid equal to 6, whereas a peer that stores two keys with values $\{63, 0\}$ has a centroid equal to 63.5, since key values are defined in a circular space, in which value 0 succeeds value $N_c - 1$. The centroid values of peers are used by agents to move the keys. The agents tend to pick a key from a peer if its value is distant from the peer centroid, and forward the key towards a peer whose centroid is as close as possible to the key value. Agents can move between consecutive peers, or can use short-cuts to jump directly to distant peers, exploiting the *finger tables* of the underlying Chord structure. These simple operations are completely decentralized, since they depend on local information only, and gradually achieve the global sorting of the keys.

Figure 1 gives an example of the way resource keys are sorted. In this sample network, peer indexes and resource keys are defined over 6 and 3 bits, respectively, and 16 peers are actually connected to the system. At the interior of the ring, the figure specifies the indexes of the peers, whereas at the exterior it reports, for every peer, the keys stored by the peer (only three keys are shown for simplicity) and the peer centroid c. It can be noted that both the values of centroids and peer indexes are sorted in clockwise direction, but they are not related to one another. Indeed, different approaches are used to sort them: the peer indexes are sorted by the Chord management operations, whereas the resource keys are sorted by the self-organizing operations of the Self-Chord agents.

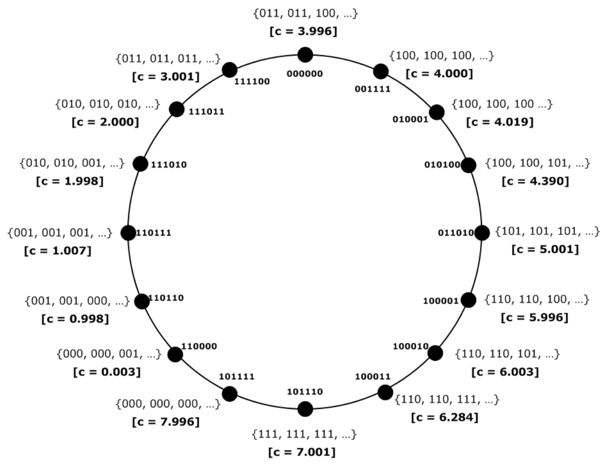

Fig. 1. Sample sorting of resource keys in the peers of Self-Chord. For each peer, its index, a number of stored keys and the centroid are reported.

2.1 Pick and Drop Operations of Self-Chord Agents

The operations of ant-inspired agents in Self-Chord are inspired by the basic ant algorithms used by Deneubourg et al. [3] to model the phenomenon of larval sorting in ants, and by Lumer and Faieta [11] to generalize this behavior to the sorting of data items. Each agent, in its lifetime, performs a few simple operations, cyclically: *(i)* while it is not carrying any key, it hops randomly from a peer to its predecessor or successor; *(ii)* at any new peer, it decides whether or not to *pick* a key with a Bernoulli trial; *(iii)* after picking a key, the agent jumps to a new peer exploiting the current peer's finger table; *(iv)* at the new peer, the agent decides whether or not to *drop* the carried key with a Bernoulli trial. Operations *(ii)* and *(iv)* are repeated until the agent picks or drops a key, respectively.

The decision about the pick operation depends on the value of the key under consideration and the centroid of the current peer. To foster the sorting of keys, it is convenient to pick keys that are distant from the peer centroid, whereas the keys that are close to it are probably already placed in the correct place. Therefore, the probability of picking a key k at a peer having centroid c is defined to be inversely proportional to the similarity between k and c. The similarity function $f(k, c)$ and the *pick* probability P_{pick} are,

$$f(k, c) = 1 - \frac{d(k, c)}{N_c/2} \tag{1}$$

$$P_{pick} = (\frac{\alpha_p}{\alpha_p + f(r, c)})^2 \quad \text{with } \alpha_p \geq 0 \tag{2}$$

where $d(k, c)$ is the distance between k and c, computed on the circular space of the keys. For example, with N_c=64, $d(12, 18.7)$=6.7 and $d(3, 63.5)$=3.5. The value of $f(k,c)$ is comprised between 0 (maximum diversity between k and c) and 1 (maximum similarity). With high probability the agent picks a key whose value is distant from the peer centroid. The parameter α_p is a threshold that can be tuned to modulate the pick probability: when $f \ll \alpha_p$, P_{pick} is close to 1; when $f \gg \alpha_p$, P_{pick} is close to 0.

Once an agent has picked a key k from a peer, it tries to jump to the region of the ring where this key should be deposited, therefore towards peers whose centroids are as close as possible to the carried key. To calculate the length of the jump, the agent exploits the fact that the peer indexes are ordered and the resource keys are also being ordered. First, the agent calculates the difference $k-c$ in the arithmetic modulo N_c, where c is the centroid of the current peer. Then, it makes a proportion between this distance, calculated in the space of resource keys, and the distance between the current peer P_s and the "destination" peer P_d, calculated in the space of peer indexes:

$$\frac{k-c}{N_c} = \frac{P_d - P_s}{N_r} \tag{3}$$

The agent tries to jump to a peer whose index is as close as possible to:

$$P_d = P_s + \frac{N_r}{N_c}(k-c) \tag{4}$$

To do this, the agent exploits the *finger table* of P_s. In Chord, the $i-$th finger of peer p, denoted by $p.finger(i)$ contains the index of the first peer, d, that succeeds the index of p by at least 2^{i-1}, namely $d = successor(p + 2^{i-1}), i = 1..B_p$. The finger table is used by Chord to let the search messages jump to distant peers, so as to complete discovery procedures in a logarithmic time, since at every jump the search space can be halved.

After calculating P_d, the agent jumps to the peer of the finger table whose index is the closest to P_d. At the new peer, the agent evaluates the *drop* operation (see the details below). If this operation is actually performed, the agent will again move towards the successor or predecessor peer, until it will pick another key. Otherwise, the agent will recalculate the value of P_d and make another jump, trying to approach better the region of the ring where the carried key should be deposited.

After each jump, the agent must decide whether or not to drop the key on this peer. The *drop* probability, P_{drop}, is,

$$P_{drop} = \left(\frac{f(k,c)}{\alpha_d + f(k,c)}\right)^2 \quad \text{with } \alpha_d \geq 0 \tag{5}$$

where k is the value of the carried key, c is the centroid of the current peer, and the similarity function $f(k,c)$ is computed as in (1). The threshold α_d has a similar meaning as α_p. Contrary to P_{pick}, P_{drop} is directly proportional to the similarity between k and c, therefore the agent tends to drop a key when it is similar to the other keys stored in the local peer.

Pick and drop operations contribute to the correct reordering of keys, because the agents tend to place every key in a peer that has a centroid value close to the key value. The performance of Self-Chord was evaluated in [6], both with respect to the effectiveness and rapidity of key sorting and to the efficiency of discovery operations. In Section 4, attention will be centered on the distance between consecutive centroids. Indeed, it is easy to see that the average distance between consecutive peer centroids, in a network with N_p peers and N_c admissible values of resource keys, must be equal to N_c/N_p. But it is also important to analyze the distribution of this distance: a narrow distribution, with many values close to the average, is an indication that the centroids are equally spaced, and a robust guarantee about the expected efficiency of discovery operations.

3 Random Matrix Theory

This section gives a short introduction to the field of random matrix theory, in particular to the properties of the Gaussian Unitary Ensemble (GUE), and to the connections with number theory. Besides the technical papers and books cited below, the reader can find an interesting summary and a critical discussion of these issues in Chapters 17 and 18 of the book “Prime Obsession” authored by John Derbyshire [4].

The random matrix theory has proved essential for the study of complex physical systems [13]. In particular, the exact calculation of the energy levels of non-trivial quantum mechanical systems is practically impossible owing to the enormous number of possible states and of the possible interactions among the involved components. However, first Wigner, and then Dyson [5], showed that a statistical description is possible by studying the properties of random matrices. In particular, the energy levels of a wide range of semi-chaotic dynamical systems are excellently fitted by the eigenvalues of the random matrices belonging to the Gaussian Unitary Ensemble (GUE), in the limit $N \longrightarrow \infty$, N being the matrix size. The GUE consists of n x n complex matrices of the form $A = (a_{jk})$, where

$$a_{jj} = \sqrt{2} \cdot \sigma_{j,j}$$

$$a_{jk} = \sigma_{j,k} + i\eta_{j,k} \quad for \quad j < k \quad and$$
$$a_{jk} = \overline{a}_{kj} = \sigma_{k,j} - i\eta_{k,j} \quad for \quad j > k$$

where $\sigma_{j,k}$ and $\eta_{j,k}$ are independent standard normal variables.

Square matrices of this kind, which are equal to their own conjugate transposes, are called *Hermitian* or *self-adjoint*, and it was proved that all their eigenvalues are real. The GUE matrices are a subset of Hermitian matrices, and for this subset a well established theory proved the existence of the so-called “repulsion effect” between consecutive eigenvalues. This means that that the probability of finding two consecutive eigenvalues spaced by an amount ω (and, correspondingly, the probability of finding two consecutive energy levels of the associated physical system spaced by the same amount) decays at small values of ω.

This property derives from the *pair correlation* function of the eigenvalues of GUE random matrices, derived by Dyson for very large matrices:

$$1 - \frac{\sin^2(\pi x)}{\pi^2 x^2} \tag{6}$$

This means, informally, that the fraction of eigenvalues couples ϕ_m and ϕ_n, for which the distance between them (i.e., the modulus of the difference of their values) is between α and β, is approximately equal to:

$$\int_\alpha^\beta (1 - \frac{\sin^2(\pi x)}{\pi^2 x^2}) dx \tag{7}$$

Notice that the value of expression (6) decreases and tends to zero for decreasing values of x, which proves the repulsion effect: the number of very close eigenvalues is much lower than the one that would be experienced if the eigenvalues were distributed uniformly.

The GUE pair correlation is also the subject of the famous Montgomery conjecture [14], which states that the distribution of the spacings between the non-trivial zeros of the Riemann Zeta function is statistically identical to the spacings of GUE eigenvalues. These non-trivial zeros, whose location in the complex plane is the subject of the famous unsolved Riemann hypothesis, are strictly related to the distribution of prime numbers. The Montgomery conjecture opened a very fruitful research stream, as physicists and number theorists do not see any reason why the Zeta function should be correlated to the behavior of a wide class of dynamical systems, among which the quantum mechanical systems mentioned above. Moreover, this unexpected correlation is deemed as one of the most promising avenues that could lead to the demonstration of the Riemann hypothesis. Following the work of Montgomery, Odlyzko [15] made an accurate analysis of the statistical properties of the zeros of the Zeta function, which consolidated the Montgomery conjecture and turned it into the Montgomery-Odlyzko law. Odlyzko compared the statistical properties of the GUE eigenvalues to those of different sets of 10000 consecutive zeros of the Riemann Zeta function.

4 On Statistical Analogies between Self-Chord Centroids and GUE Eigenvalues

As mentioned in Section 2, the effectiveness of the Self-Chord P2P system is strongly related to the distribution of the spacings between consecutive peer centroids. Several hints suggest a possible analogy between this distribution and the distribution of GUE eigenvalues. They are:

1. the processes modeled by GUE matrices have a random and semi-chaotic nature. The key sorting in Self-Chord is a random process and its behavior has also semi-chaotic characteristics: after an external perturbation, the key sorting is recovered, but the value of a single peer centroid is almost unpredictable;

2. in Self-Chord there is a clear repulsion effect between consecutive centroids: since the ants continuously sort the keys with respect to centroids, the centroids tend to keep a minimum distance between each other. Similarly, large spacings are also unlikely. In general, the values of centroid spacings tend to be close to the average, much more than what would be experienced if the centroid values were not biased, i.e., if they followed a Poisson distribution. A similar phenomenon is experienced by consecutive GUE eigenvalues;
3. the most popular application of GUE operators concerns the study of the energy levels of quantum mechanical systems. These systems have a strong connection with number theory, because energy levels derive from the composition of the energy states of single components, which are not defined on a continuum but must be multiple of energy *quanta*. In the same fashion, the value of a centroid is calculated from the values of a finite number of keys, and the keys can only have integer values;
4. continuing the comparison to particle systems, the energy levels may change their values after the absorption, or the emission, of energy quanta. These modifications happen continuously under the effect of inner statistical processes and external perturbations. A similar phenomenon happens in the Self-Chord system, owing to the pick and drop operations performed by ants as the result of random Bernoulli trials. After a couple of pick/drop operations, a key with an integer value is moved between two peers, and the respective centroids change their values accordingly.

Of course, these are only hints, but, quite surprisingly, they were confirmed by a large number of statistical experiments. The distribution of centroid spacings was calculated in Self-Chord systems where: the number of peers N_p is equal to 2000, 5000 and 10000 peers; the number of admissible peer indexes is $N_c = 1,000,000$; each peer published 100 keys on average; the values of thresholds α_p and α_d are set, respectively, to 0.3 and 0.4. Experiments were performed using the Self-Chord event-based simulator available at the Web site http://self-chord.icar.cnr.it. The centroid distribution was compared to the distribution of GUE eigenvalues, and the result is reported in Figure 2. To obtain the distribution of centroid spacings, a histogram was calculated for values of spacings between 0 and 4 (the average was normalized to 1), and step 0.04. The distribution of GUE spacings, suitably normalized, can be obtained through a quite complex mathematical procedure devised by Mehta and des Cloizeaux, and detailed in [12]. For the purpose of this work, the distribution was provided directly by Odlyzko, whose landmark work was mentioned in the previous section.

The two distributions are clearly very similar. The zigzag deviations of the centroid distributions may be caused by the limited size of the system. In fact, deviations become smaller as the system size increases, as noticed in the figure, which seems to mean that the contribution of the hypothetical white noise component tends to vanish in very large systems. This can be seen as a further confirmation about the correctness of the alleged similarity: the theoretical distribution of GUE eigenvalues is related to matrices whose size tends to infinity. It should be noticed that the GUE distribution is remarkably different from any

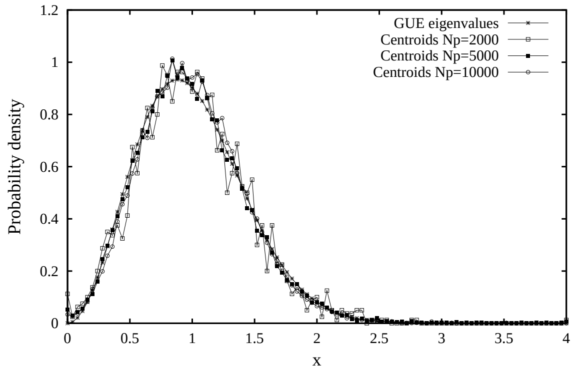

Fig. 2. Probability density of normalized spacings between consecutive centroids, computed for Self-Chord networks with different size, compared to the distribution of normalized spacings between GUE eigenvalues.

common statistical distribution - normal, truncated normal, Gamma, Poisson, etc. - and that neither the centroid spacings can be approximated by any of these distributions.

Figure 2 is useful to visibly compare the distributions, but cannot be considered as a stringent statistical test. The “q-q plot”, or quantile-quantile plot, is a more rigorous test, very useful to detect the deviations between a theoretical distribution (in this case, the GUE distribution) and an empirical one (the centroid spacings). This test was used by Odlyzko for his comparison between the GUE eigenvalues and the zeros of the Zeta function. The q-q graph plots the quantiles of two distributions one against the other. A quantile is defined as the value of the random variable under observation that is strictly greater than a given fraction of samples. For example, let us consider the point of the q-q plot that corresponds to the 0.3 quantile: the x coordinate of this point is the value at which the GUE cumulative distribution is equal to 0.3, whereas the y coordinate is the value that is higher than 30% of the values of the centroid spacings.

Figure 3 shows the q-q plot for different network sizes, calculated at the quantiles 0.02, 0.04, up to 0.98. It appears that the three graphs are all very close to the straight line $y = x$, which would be obtained if the theoretical and empirical distributions were identical. The largest deviations are observed at the two ends of the plot. This is not surprising, since the first and the last quantiles are related to the tails of the distribution. They are therefore determined by the sample values that are the most distant from the average, and are particularly subject to statistical perturbations. But, again, these deviations seem to wane as the network size increases.

An interesting question is whether the centroid distribution depends on the particular setting of Self-Chord parameters. Actually, there are only two parameters, the α_p and α_d thresholds used for pick and drop operations. If the

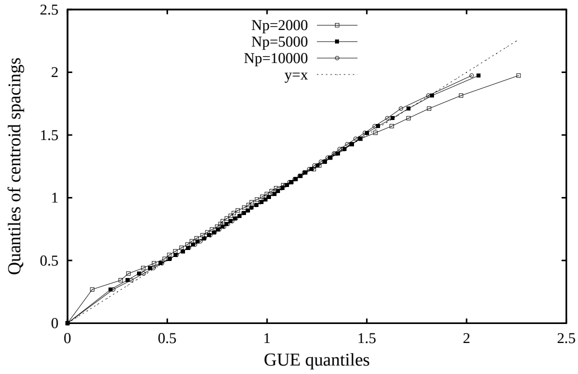

Fig. 3. Q-q plot of the centroid distributions, plotted against the theoretical GUE distribution. The straight line y=x is plotted for comparison.

discovered similarity with the GUE distribution were valid only for a particular setting, the phenomenon could not be considered general. Fortunately, this is not the case. A number of experiments were performed, in networks with 2000 peers, when varying the value of each of the two parameters, with the other kept constant. Figure 4 reports the obtained values of the Pearson coefficient, which summarizes the correlation between centroid and GUE distributions with a single value. This statistical index can assume values between -1.0 and 1.0, and the latter value corresponds to a linear correlation between the two compared distributions. The figure shows that the Pearson coefficient is greater than 0.97 for wide ranges of values of α_p and α_d. Indeed, these are the values that allow Self-Chord to sort the keys properly. In other words, with any setting that ensures a successful sorting of keys, the distribution of centroids is always very similar to the GUE distribution.

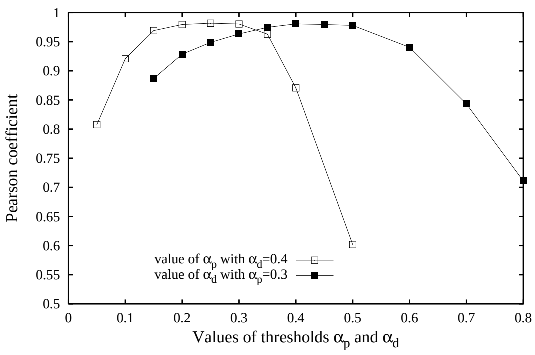

Fig. 4. Pearson coefficient measuring the correlation between centroid and GUE distributions. Values of the coefficient are reported when varying the value of either α_p or α_d.

5 Conclusion

What conclusions can be drawn from the results reported in the previous section? First of all, they offer a robust statistical confirmation about the Self-Chord capacity of sorting the peer centroids and limiting the variations of the spacings among them: indeed, the observed distribution of spacings is quite narrow, and high values of spacings are extremely unlikely. This phenomenon, along with the observation that the keys stored by a peer are always very similar to the local centroid [6], ensures that a quest for a key can be turned into the quest for a peer centroid, and that a given centroid can be discovered in logarithmic time with a high level of guarantee. This is an interesting result *per se*, as it confirms that ant algorithms can be used to give self-* properties to structured P2P systems, which has been deemed impossible for a long time. It should be noticed that algorithms of this kind can be devised not only for Chord, but also for other popular P2P systems, like CAN and Pastry.

Now, let us turn to the questions asked in the introductory section. Firstly, is this phenomenon tied to the P2P scenario, or is it valid for a wide class of ant algorithms? The algorithm used in Self-Chord is very similar to many basic ant algorithms, for example to those described in Chapter 4 of the renowned book on Swarm Intelligence authored by Bonabeau et al. [1]. Therefore, it is plausible that this behavior applies to ant algorithms in general. If this is true, is it possible to derive a GUE operator (i.e., an operator that uses GUE matrices) that models the Self-Chord process, or other processes based on the ant paradigm? This would be particularly important, since today there is a significant need for rigorous methodologies that may help to design and analyze ant algorithms. It should be considered that ant algorithms are widely exploited to solve a large number of complex problems, such as task allocation, routing problems, graph partitioning, etc., but the detailed behavior of these algorithms is still mostly obscure.

Finally, there is the not yet unveiled tie between random matrix theory and the Riemann Zeta function. Is it conceivable that ant algorithms are correlated to the theory of prime numbers? Could this correlation concern only artificial ants or even real ants? No, this is too much beyond any reasonable conjecture.

Acknowledgments. I would like to thank Agostino Forestiero and Michela Meo for their contribution to the Self-Chord project. I am also grateful to John Derbyshire, Andrew Odlyzko and Emanuele Viterbo for their useful advices.

References

1. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems.* Oxford University Press, New York, NY, USA, 1999.
2. Amos Brocco, Fulvio Frapolli, and Béat Hirsbrunner. Blåtant: Bounding networks' diameter with a collaborative distributed algorithm. In *Proc. of the 6th InternationalConference on Ant Colony Optimization and Swarm Intelligence, ANTS 2008*, pages 275–282, Brussels, Belgium, 2008.

3. J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting: robot-like ants and ant-like robots. In *Proc. of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 356–363, Cambridge, MA, USA, 1990. MIT Press.
4. John Derbyshire. *Prime obsession: Bernhard Riemann and the greatest unsolved problem in mathematics*. Joseph Henry Press, 2003.
5. F. J. Dyson. Statistical theory of the energy levels of complex systems. III. *Journal of Mathematical Physics*, 3(1):166–175, February 1962.
6. Agostino Forestiero, Emilio Leonardi, Carlo Mastroianni, and Michela Meo. Self-chord: a bio-inspired p2p framework for self-organizing distributed systems. *IEEE/ACM Transactions on Networking*, 2010. Published online, DOI: 10.1109/TNET.2010.2046745.
7. Agostino Forestiero and Carlo Mastroianni. A swarm algorithm for a self-structured P2P information system. *IEEE Transactions on Evolutionary Computation*, 13(4):681–694, August 2009.
8. Agostino Forestiero, Carlo Mastroianni, and Michela Meo. Self-chord: A bio-inspired algorithm for structured p2p systems. In *CCGRID '09: Proc. of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 44–51, Shanghai, China, 2009.
9. Christophe Guéret, Nicolas Monmarché, and Mohamed Slimane. A biology-inspired model for the automatic dissemination of information in p2p networks. *Multiagent and Grid Systems*, 3(1):87–104, 2007.
10. Steven Y. Ko, Indranil Gupta, and Yookyung Jo. A new class of nature-inspired algorithms for self-adaptive peer-to-peer computing. *ACM Transactions on Autonomous and Adaptive Systems*, 3(3):1–34, 2008.
11. Erik D. Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proc. of SAB94, 3rd international conference on Simulation of adaptive behavior: from animals to animats 3*, pages 501–508, Cambridge, MA, USA, 1994. MIT Press.
12. M. L. Mehta and J. Cloizeaux. The probability of several consecutive eigenvalues of a random matrix. *Ind. J. Pure Appl. Math.*, 3:329–351, 1972.
13. Madan Lal Mehta. *Random Matrices*. Academic Press Inc., Boston, MA, USA, 1991.
14. H. L. Montgomery. The pair correlation of zeros of the zeta function. pages 181–193, Providence, RI, USA, 1973.
15. Andrew Odlyzko. On the distribution of spacings between zeros of the zeta function. *Mathematics of Computation*, 48(177):273–308, 1987.
16. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of the Conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM'01*, 2001.
17. Katia Sycara. Multiagent systems. *Artificial Intelligence Magazine*, 10(2):79–93, 1998.