

Document downloaded from:

<http://hdl.handle.net/10251/61320>

This paper must be cited as:

Krynicky, K.; Jaén Martínez, FJ.; Mocholi Agües, JA. (2014). Ant colony optimisation for resource searching in dynamic peer-to-peer grids. *International Journal of Bio-Inspired Computation*. 6(3):153-165. doi:10.1504/IJBIC.2014.062634.



The final publication is available at

<http://dx.doi.org/10.1504/IJBIC.2014.062634>

Copyright Inderscience

Additional Information

Ant colony optimization for resource searching in dynamic peer-to-peer grids

To appear at International Journal of Bio-Inspired Computation

<http://www.inderscience.com/jhome.php?jcode=ijbic>

DOI: Pending

Kamil Krynicki, Javier Jaen, Jose A. Mocholi

ISSI - Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València,
Camino de Vera, s/n, 46022 Valencia, Spain
{kkrynicki, fjaen, jmocholi}@dsic.upv.es

Corresponding author:

Kamil Krynicki

Camino de Vera, s/n, 46022 Valencia, Spain

Email: kkrynicki@dsic.upv.es

Phone: +34 96 387 35 69

Fax: +34 96 387 73 59

Abstract

The applicability of peer-to-peer (p2p) in the domain of grid computing has been an important subject over the past years. Nevertheless, the sole merger between p2p and the concept of grid is not sufficient to guarantee nontrivial efficiency. Some claim that ant colony optimization (ACO) algorithms might provide a definite answer to this question. However, the use of ACO in grid networks causes several problems. The first and foremost stems out of the fact that ACO algorithms usually perform well under the conditions of static networks, solving predetermined problems in a known and bound space. The question that remains to be answered is whether the evolutive component of these algorithms is able to cope with changing conditions; and by those we mean changes both in the positive sense, such as the appearance of new resources, but also in the negative sense, such as the disappearance or failure of fragments of the network. In this paper we study these considerations in depth, bearing in mind the specificity of the peer-to-peer nature.

Keywords

Semantic Search, Ant Colony Optimization, Peer-to-peer, Grid computing, P2P Grid

1 Introduction

The idea of the grid originated in the early nineties as a new metaphore for distributed computing with resources that are heterogeneous and dynamic in nature. The key idea behind it was to loosely couple computing nodes to allow them to collaborate in order to achieve a common goal. Due to an apparent ambiguity and similarity to other solutions such as parallel computing and computer clusters a proper definition was provided in Foster (2004). A checklist of requirements that define grid computing consists of three primary attributes: computing resources are not administered centrally, open standards are used, nontrivial quality of service is achieved.

Over the past years many subgenres of grid computation have spawn. They include many concepts, just to name a few: Software as a Service OpenCrowd (2011), DataGrids Hoschek et al. (2000), CPU scavenging – with its prime example Seti@Home (1999). The last mentioned is, interestingly, also a very good example of peer-to-peer (p2p) cooperation and a proof of these two computation architectures complementing each other. Trunfio et al. (2007)

formalize the split between the technologies stating that the grid used to be restricted to scientific and limited applications as opposed to p2p which was much more wide scale and accessible.

The key point of the grid computing model is the question of the underlying protocol and node organization. One of the most prominent approaches is the use of p2p architectures as a base building block (the communication layer) and authors agree that it is a suitable platform for grid computing Talia and Trunfio (2003), sometimes seeing it as a sister technology. This distributed infrastructure does fulfil automatically the first two of the three required points – the distributed resources and open standards. Also, it allows transferring all the knowledge developed in the field of p2p with all its benefits to the grid computing. The third point, however, which is the question of the efficient service remains to be answered, which becomes even more crucial if we take into consideration the growing scope and highly dynamic nature of p2p networks.

Among the existing biologically plausible existing approaches, Parpinelli and Lopes (2012), Xiao and Chen (2013) and Mozaffari et al. (2012), in this work we chose to examine the ant colony optimization metaheuristic as a possible solution to the effective searching of resources in dynamic p2p grids. ACO algorithms achieved very good results in problems that include routing and resource discovery as shown in Dorigo and Gambardella (1997), and there were even attempts, of limited extent, to apply them to the p2p grid itself Deng et al. (2009), which proves the importance of this idea. However, no previous work has analysed the suitability of this algorithmic strategy under all the types of dynamism considered in our study.

Consequently, in order to observe and analyse the applicability of ACO under those conditions we have designed several experiments to isolate a set of real-life resembling forms of variability that a network might experience and, consequently, subject selected algorithms to tests based around those forms of variability. The direct objective is to establish whether or not the algorithms designed to work in static environments can cope with various forms of dynamism.

2 Related work

The related works, to the question of resource discovery in general, might be categorized as follows: the centric solutions, the grid solutions, the p2p Grid solutions and the p2p Grid solutions with ACO as the search algorithm.

The centric solutions are feasible only in small-scale due to a strong limitation cap of processing power, storage space, etc. Another drawback is that servers are single points of failure, creating potentially vulnerable systems. Consequently, there have been steps taken in order to disperse the system among a grid of nodes – following the metaphor of an electric power grid Foster (2004), with many challenges of its own. Some try to achieve efficiency in such setups using proprietary or mixed approaches Han and Youn (2009), Brocco et al. (2010), some base themselves on the p2p paradigm Di Stefano et al (2009), while other use ACO algorithms or ACO-grid combinations Dhillon and Van Mieghem (2007), Forestiero et al. (2008).

In the mixed approaches category it is worth pointing out the work of Brocco et al. (2010) presenting an interesting approach to the subject of resource discovery in grid networks. They prepared a hybrid solution encompassing a wide range of algorithms: ACO, local flooding with caching and replication based on gossiping algorithms; they report satisfactory results, but limiting the size of the network to 1281 nodes and analysing their algorithm under just one type of dynamism, defined in this paper as reconfiguration, which is just one of the types of dynamism considered in our work. A purer approach is presented in Forestiero et al. (2008), where an ACO-like mechanism is the only one that governs the behaviour of search agents in a grid. Here, similarly, the concept of data descriptors' replication is present, which is outside our objectives and, as it occurs in Brocco et al. (2010), only reconfiguration operations are considered as a form of dynamism affecting the underlying network.

We must also mention here the work of Deng et al. (2009). They touched upon the subject of ACO in p2p Grids, however with slightly different emphasis. In our case the experimental study focus is placed on ACO, in their work it is on p2p Grids. Also, while we examine various existing ACO algorithms and observe their behaviour when applied to the question of p2p, they design a proprietary ACO algorithm (roughly based on ACS Dorigo and Gambardella (1997)) along with p2p Grid and test its robustness against a non-ACO method, namely distributed hash table routing. The scope of our experiments is also larger, extending to thousands of nodes and tens of thousands of links over one hundred thousand iterations. Additionally they do not take in consideration the fundamental concept of dynamism in the network as we do.

In summary, the main contribution of this work is twofold. First, it is a pure ACO solution in the field of p2p Grids, without any kind of hybridization in form of replication, flooding etc. Second, we attempt to exhaust the subject of structural dynamism and its impact on ACO in general – a subject that is very seldom explored to this extent. We do so in networks, which are built to be as close as possible approximations of real-life systems – they consist of thousands of nodes and tens of thousands of links.

3 Ant colony optimization in p2p

Ant Colony Optimization Colom et al. (1991), Dorigo and Di Caro (1999) is a swarm intelligence approach to problem-solving introduced by Marco Dorigo in his work on distributed optimization in 1991. The core idea of ACO is twofold, firstly, as properly named, it uses a swarm of simple and stochastic automaton to solve complex problems and, secondly, the communication between these is through stigmergy and therefore indirect. Such a communication method has shown to provide interesting results, especially with the emphasis on finding the shortest path Goss et al. (1989), optimizing a given function Mocholi et al. (2012a), Mocholi et al. (2012b) or in other graph-related problems Aicha et al. (2010), Abdelaziz et al (2012) and Ahangarikiasari et al. (2013). The automaton, or agents, in ACO are called ants. Each ant has the simple task of finding the required resource (search phase) and bringing it back to its nest (returning phase); without the loss of generality one can limit the world, in which ants live, to a bidirectional graph $G(V, E)$ of finite size with vertices $v \in V$ representing possible locations of resources and edges $e \in E$ representing trails.

In order to consider ACO p2p-compliant we need to formulate additional requisites. Every ACO-based query resolution algorithm in p2p environments must conform to the below query-resource (q-r) principles:

1. Every node may have any amount of resources – including zero resources.
2. Every node may issue a query – that is, a request for a set of resources of any nature; one that may be constructed of resources residing in one or many nodes within the network.
3. Every node must not be aware of the content of any other node but itself.
4. Every node may be connected to a set of nodes via bidirectional links of high traveling cost. A Degenerated (disconnected) node may be connected to zero other nodes.
5. Every query is propagated among nodes, collecting resources that correspond to the request issued.
6. The destination (the final) node of a query is never known a priori nor is it deterministic.
7. The trail of a query is never known a priori nor is it deterministic.

ACO methods can be grouped into two broad groups, namely: single ant and multi ant, with the distinction revolving around the amount of ants generated for solving one act of search. For each group we chose a representative for our experimental evaluation: RC-ACS, our proposed semantic extension of ACS Dorigo and Gambardella (1997) and Semant Michlmayr (2006a) for single- and multi- ant approaches respectively. These two representatives have been selected after a prior examination of ACO algorithms based on two factors: firstly, compliance with the previous (q-r) principles, i.e., ACO strategies violating any of the previous q-r principles were discarded because they would not be applicable in the p2p domain and, secondly, recognized performance under static conditions.

Our semantic extension of ACS, RC-ACS, which is based on the pheromone-per-keyword heuristic as in Michlmayr (2006a), is designed taking into account that in a network p2p there is a large quantity of traffic of very different natures. The pheromone left by queries for certain resources must not influence the search of other, which may be unrelated. In order to tackle this problem we introduce the notion of Routing Concept, noted as ρ . It requires that every edge maintains, not a single pheromone value $\tau(vu)$ between its source node v and each neighbour node u , but rather an associative array $\tau_\rho(vu)$, where ρ is obtained from the current request by means of $\rho = \rho(Q)$, where $\rho(Q)$ is a function that extracts the Routing Concept based on the Query Q issued. In our case we defined the function $\rho(Q)$ as one that returns the taxonomical label of the parent concept of the requested one.

Ants follow a simple and non-deterministic search algorithm that we summarized in Figure 3-1.

4 Experimental methodology

We define the graph topology as follows: the graph $G(V, E)$ consists of $|V|$ vertices, where $|V|$ is an even number. The edges E are organized in a fully connected grid with a toroidal topology, where both dimensions d_1, d_2 of the creating rectangle are chosen to fulfil $|d_2 - d_1| \cong 0$ to minimize the medium distance. Consequently we create the graph $G'(V, E \cup LDC)$, where LDC is a set of all long distance connections (LDC_{ij}) in the graph G ; every vertex v_i

has exactly one long distance edge (LDC_{ij}) that connects it directly to a random vertex v_j , with the pure toroidal distance, that is - using only edges in E , $d_E(v_i v_j) > 2$.

$$LDC \subset \{v_i v_j | (v_i v_j) \notin E \wedge d_E(v_i v_j) > 2\}$$

The probability of a node v_i having LDC_{ij} of length $d_E(v_i v_j)$ is proportional to $d_E(v_i v_j)^{-1}$. The above explanation is a strict reformulation of the approach from Michlmayr (2006a).

Although the execution of every experiment will begin with the G' topology, this topology will be subject to changes according to the type of dynamism studied in each experiment. Therefore, each experiment will define two concepts:

1. The modification function of the given graph $\Delta: G \rightarrow G$, which transforms the graph G
2. The modification function period Δ_p , is an integer which expresses how often Δ function is executed.

Moreover, each experiment will be evaluated according to a common quality measure called Hop per Hit (HpH), which is dimensionless ratio that has been effectively used to measure performance in p2p systems in previous works Michlmayr (2006a), Krynicki et al. (2013). In this metric hop is the total amount of steps taken by an agent and hit is the amount of resources found.

See Table 1 for details on algorithm execution parameters.

4.1 Test Setup

Every test will consist of an amount it_{max} of iterations. Iteration is defined as a single act of querying, launched from a random node, routed according to the given algorithm. For every query a set of data will be stored: the birth (creation) nanosecond, the death nanosecond, the query as text, the number of hops made, the number of resources found and the location of resources found. Due to the large amount of data and its high variability, we chose to use simple rolling average of the size 254 as an impulse filter and plot the function $HpH(it)$, where it is the iteration number. The amount of queries will be fixed at $it_{max} = 100000$.

Our testing platform is a highly configurable Java-based engine that supports all the above algorithms. Tests will be run on Intel Pentium 4 630 at 3.00GHz with 4 GB of ram on a 32bit Windows 7 machine.

4.1.1 Taxonomy and resource distribution

ACM Computing Classification System ACM (1998) will be the taxonomical vocabulary used. Every resource in the network G' is described by one, and only one leaf taxonomical concept t of the ACM classification. A resource has therefore only two properties: its owner vertex v and a taxonomical label t . It is written as $r(v, t)$. Note that $v_1 = v_2 \wedge t_1 = t_2 \not\Rightarrow r_1(v_1, t_1) = r_2(v_2, t_2)$. Such an approach leads to valuing higher those nodes that provide many resources of the same t , which is the objective.

The distribution of resources within the network follows strictly the approach Michlmayr (2006b). The resources are evenly distributed among the nodes, as well as among the entities in the taxonomy tree. Additionally, every node is a designated expert in a given field (there can be multiple experts in each field) which is expressed by the composition of resources in it. Of all the resources units in a node, 60% is labelled with the field in which the node is considered an "expert", further 20% is labelled with another field that is closely related in the taxonomical tree to the expert field, and the last 20% is purely random, but with the restriction to be outside the expert field. This is said to resemble real-world distribution more, reflecting the fact that people have specific interests and hobbies.

4.1.2 Query and query resolution

Every query q will only carry one of the ACM classification leaf entities and it will be fully defined by it. In this case, however, $q_1(t_1) = q_2(t_2) \Leftrightarrow t_1 = t_2$. The benefit of such an approach is to be able to compare results of two queries released at different time points in the testing process and to show relative improvement between them.

The resolution of a query $q_r(t_r)$ in a vertex v consists of finding all the resources that have been labeled with t_r , that is, all the resources $r_r \in \{r | \exists r(v, t_r)\}$.

During the evaluation process every node of the graph G of size $|E|$ has a probability of being chosen to generate a query q with the probability of $1/|E|$.

4.2 Types of dynamism

We decided to focus on the most common types of dynamism a real-life p2p network can experience. They are: expansion (the process of densification of the network), contraction (the process of rarefaction of the network), enrichment (the process of addition of new resources) and reconfiguration (the process of migration of nodes within the network) as it is summarized in Table 2.

The network expansion is one of the most straightforward ways to improve the robustness of algorithms within a p2p network. The subject of expanding the network by adding links in both uninformed and informed manner has been examined many times, for instance in the case of Overlay Networks and Interest-Based Locality, as well as Acquaintance Links. Here we will show the effect of random network growth specifically on ACO algorithms in order to express the fact that users might be allowed to create direct links, or shortcuts, to their favourite nodes, with the assumption that it is somewhat a random process.

The network contraction is a far less studied phenomenon. In this case the nodes and their corresponding resources remain unchanged, but the interconnecting network does shrink, leaving nodes disconnected and the system partitioned. Naturally this must hinder the algorithms ability to obtain good results therefore the response should be negative. This kind of behaviour might be seen as a situation in which the communication between the nodes is progressively worsened by, for instance, an on-going electrical storm.

By the network enrichment we understand the increase in the amount of resources that every node provides. This clearly must enable the algorithms to gather more resources, but the question of whether the newly provided resources can be effectively discovered remains to be answered. Note that it is much more likely for users to add resources than to remove them so the phenomenon of resources massively disappearing from the system is secondary. Hence the counter-part to this dynamism, might we call it “Network impoverished”, is not included. This is a direct conclusion from a universal and long-lasting trend of ever-increasing traffic over file sharing services and similar. The process of network enrichment from the perspective of the processing grids could be seen as an act of expanding the computer capabilities to provide more services.

The network reconfiguration is the more important process, the more mobile the network’s users are. If a p2p network provides location based resources one would suppose it would be beneficial to reattach the user to the nodes in the current neighbourhood. This directly implies a disappearance from the original location and appearance in another. It is a very interesting concept to examine from the ACO’s point of view, seeing how they take a period of time to stabilize a path and then to remove it. If a key resource provider in a given neighbourhood disappears it should lead to a sudden drop in the quality of results therefore we expect this type of dynamism hinder results severely.

5 Experimental Study

In this section we present all the experimental results. In experiments 1 - 4 we examine in detail all the types of dynamism mentioned in section 4.2. In every experiment the RandomWalks k2 is shown as the baseline behaviour and the minimum HpH expectancy.

5.1 Experiment 1: NetGrow 10k

We have executed this experiment in 6 independent configurations; see Table 3 for the details.

The most visible conclusion from the results presented in Figure 5-1 is that both algorithms struggle to appreciate the additional shortcuts added to the system. In case of multi-ant behaviour it is particularly striking, as there is absolutely no improvement detected in terms of HpH. Moreover, SemAnt remains better than random behaviour only by a slight margin. There is improvement detected in case of RC-ACS, but not proportional to the amount of edges added, as in the variant NetGrow10k 480links we increase the edge set from 2560 to 6880, while the final average HpH reaches only 20% improvement.

5.2 Experiment 2: ResGrow 10k

As in real life resources constantly appear and disappear from the system; if you consider data grid systems, replication services may constantly add new files to the distributed repositories. Therefore, the question to answer here is how effective are the proposed algorithms in finding emerging resources, while the already found ones are still

in place. In order to simulate this dynamic we designed ResGrow experiment. See Table 4 for the independent experiment configurations used.

This experiment was designed as the one with the most positive impact expected and, also, the one that affects the given state of pheromone the least, as it only manipulates the resources. Consequently, it was expected that, in the worst case, the results obtained are at least as good as the alteration-free execution.

As it can be observed in Figure 5-2 this indeed was the case. Every algorithm reported a significant improvement, disproportionally large, when compared to the increase of the amount of resources per node – in the most extreme case the improvement on HpH measure reaches 28% as a result of the growth of the resource pool by only 13%. Note that in Figure 5-2 the HpH improves linearly with the total amount of resources distributed, as well as with the amount of resources per growth. This means that all the algorithms are capable of finding new resources; much better than new routes, as shown in the Experiments 1. Note that even the random baseline improves with time. Thanks to its multi-ant penetration capabilities, Semant is the most efficient in the HpH measure in terms of relative improvement; still remains inferior in absolute terms, however.

5.3 Experiment 3: NodeMigrate 10k

This experiment tries to recreate the conditions of a live and dynamic network of nodes. The real-life analogue of the NodeMigrate can be understood as, for instance, mobile phone based p2p network where users constantly change location. With the change of location comes the process of reattachment to the network, yet the owner's node will provide the same resources as before. Another example of this situation would be the migration of files in a data grid replication service as a consequence of changing QoS characteristics related to the network or certain computing nodes. The question here is whether the evaluated algorithms are capable of erasing the path that stopped providing resources.

The results obtained from this experiment confirmed precisely our forecasts. In this case the task before the algorithms was more difficult because in this situation they had to, not only, find a new source of resources, but also forget the already established paths. The alteration of the network is increasingly impacting until nearly full network reboot in the case of 90% migration.

For multi-ant ACO b) is a perfect depiction of the actual struggle. After the transformation of the network there is an eruption of ants that try to seed new pheromone trails and, naturally, the larger the migration the more ants appear. In terms of absolute efficiency (Figure 5-3) there is a noticeable drop, but interestingly, it is higher in case of RC-ACS than in case of Semant. RC-ACS still obtains better overall results: under 20 HpH, while Semant above 20 HpH. Due to the fact that the overall structure was maintained, as there was no link removal or addition as a result of the migration, the Random behaviour was largely unaffected by the entire process.

5.4 Experiment 4: NetShrink 10k

This experiment is the most impacting of the two negative experiments due to the fact that here links, and consequently the resources, disappear irreversibly. In Table 6 we placed all the execution variants, note how in the most extreme case NetShrink10k 480links there will be no links in the system after the experiments concludes. Bear in mind that we expected an increasingly large portion of queries to become unresolved ($hit = 0$) in this setup which will simply result in lack of data points and somewhat reduced possibilities of comparison of the HpH measure. To address this issue we decided, instead of dropping the unresolved data point, to penalize it with the value 25 HpH. This caused the HpH graph for this particular experiment to be incomparable with the other, from the experiments 1 - 3.

The first observation that stands out is the relatively low impact the amount of links has. Even though it has been stated on several occasions Michlmayr (2006b) it still is interesting to see that reaching $\overline{deg(v)} < 3.00$ does not impair the HpH value much. RC-ACS in the network with $\overline{deg(v)} = 0.78$, which indicates isolated nodes, is better than Semant with $\overline{deg(v)} = 4.82$. Note that the final sections of the graph for NetShrink10k 480links, when there are no links at all, still do not converge to the value 25 due to node's capability of answering themselves the asked query.

As shown in Figure 5-4 all the algorithms are mostly unaffected with the exception of the two most impacting variants: NetShrink10k 240links and NetShrink10k 480links; the former however only slightly. As a result of this experiment we may conclude that the design of a lean network structure is important, with shortcuts added only in an educated way, in order to optimize the benefit. Moreover, links can be dropped from the system with ongoing ACO convergence process without much harm.

6 Conclusions and Future Work

In the introduction we have stated a question in regard to the applicability of ACO algorithms to the problem of effective resource searching in dynamic grid computing environments and the concept of peer-to-peer connectivity in general. Our main focus was the dynamism of networks and its implications to the convergence process of ACO algorithms.

In both the NetGrow experiments our semantic extension of ACS, RC-ACS, behaved the best, being able to use the newly added links most effectively. In the ResGrow experiment RC-ACS maintained the best overall effectiveness. In negative impact experiments (NetShrink, NodeMigrate) RC-ACS lost the least of its initial effectiveness and came out ahead yet again. It is the most telling in case of NodeMigrate, which is, at once, the most natural systems' behaviour and the most negatively impacting.

The main question asked here was whether ACO could be successfully applied in the field of grid computing and p2p in general and the answer is: it can, but only when chosen carefully while following several restrictions, as defined in section 3. Moreover, our proposed single ant semantic extension of ACS has exhibited better performance than Semant, the existing multi-ant reference work in the field of ACO semantic search. The future work will consist of further analysis of various dimensions on the robustness and efficiency and the creation of a new ACO algorithm, which would encompass all the desirable aspects to cope with different forms of dynamism.

7 Acknowledgements

This work was funded by the Spanish Ministry of Education and Science and Innovation under the National Strategic Program of Scientific Research, Development and Technological Innovation (I+D+i) and project TIN2010-20488. Kamil Krynicki is supported by the FPI fellowship from Universitat Politècnica de València.

References

- Foster, I. (2004) *What is the Grid? A Three Point Checklist*, University of Chicago, Chicago
- OpenCrowd (2011) Software as a Service (SaaS). <http://cloudtaxonomy.opencrowd.com/taxonomy/software-as-a-service/>
- Hoschek, W., Jaen, J., Samar, A., Stockinger, H. and Stockinger, K. (2000) Data Management in an International Data Grid Project, in *Proc. First IEEE/ACM International Workshop on Grid Computing GRID '00*, London, UK, pp. 77 - 90.
- Seti@Home (1999) University of California. <http://setiathome.berkeley.edu/>
- Trunfio, P., Talia, D., Papadakib, H., Fragopoulou, P., Mordacchini, M., Pennanen, M., Popov, K., Vlassov, V. and Haridi, S. (2007) Peer-to-Peer resource discovery in Grids: Models and systems, *Future Generation Computer Systems*, Vol. 23, No. 7, pp. 864-878.
- Talia, D. and Trunfio, P. (2003) Toward a Synergy Between P2P and Grids, *IEEE Internet Computing*, Vol. 7, No. 4, pp. 95-96.
- Parpinelli, R.S., Lopes, H.S. (2012) Biological Plausibility in Optimisation: an Ecosystemic View, *International Journal of Bio-Inspired Computation*, Vol.4, No.6, pp.345 - 358
- Xiao, R., Chen, T. (2013) Relationships of Swarm Intelligence and Artificial Immune System, *International Journal of Bio-Inspired Computation*, Vol.5, No.1, pp.35 - 51
- Mozaffari, A., Fathi, A., Behzadipour, S. (2012) The great salmon run: a novel bio-inspired algorithm for artificial system design and optimisation, *International Journal of Bio-Inspired Computation*, Vol. 4, No.5, pp. 286 - 301
- Dorigo, M. and Gambardella, L. M. (1997) Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary*, No. 1, pp. 53-66
- Deng, Y., Wang, F. and Ciura, A. (2009) Ant colony optimization inspired resource discovery in P2P Grid systems, *The Journal of Supercomputing*, Vol. 49, No. 1, pp. 4 - 21

- Han, Y. and Youn, CH. (2009) A new grid resource management mechanism with resource-aware policy administrator for SLA-constrained applications, *Future Generation Computer Systems*, Vol. 25, No. 7, pp. 768-778
- Brocco, A., Malatras, A. and Hirsbrunner, B. (2010) Enabling efficient information discovery in a self-structured grid, *Future Generation Computer Systems*, Vol. 26, No. 6, pp. 838-846
- Di Stefano, A., Morana, G. and Zito, D. (2009) A P2P strategy for QoS discovery and SLA negotiation in Grid environment, *Future Generation Computer Systems*, Vol. 25, No. 8, pp. 862-875
- Dhillon, SS. and Van Mieghem, P. (2007) Performance analysis of the AntNet algorithm, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 51, No. 8, pp. 2104-2125
- Forestiero, A., Mastroianni, C. and Spezzano, G. (2008) QoS-based dissemination of content in Grids, *Future Generation Computer Systems*, Vol. 24, No. 3, pp. 235-244.
- Colom, A., Dorigo, M. and Maniezzo, V. (1991) Distributed Optimization by Ant Colonies, in *Première Conférence Européenne Sur la Vie Artificielle*, Paris, France, pp. 134-142.
- Goss, S., Aron, S., Deneubourg, J.L. and Pasteels, J.M. (1989) Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, Vol. 76, pp. 579-581
- Mocholí, J.A., Jaen, J., Krynicki, K., Catalá, A., Picón, A. and Cadenas, A. (2012) Learning semantically-annotated routes for context-aware recommendations on map navigation systems. *Applied Soft Computing* Vol. 12, No. 9, pp. 3088-3098
- Mocholí, J. A., Martinez, V., Jaén, J. and Catalá, A. (2012) A multicriteria ant colony algorithm for generating music playlists, *Expert Systems with Applications*, Vol. 39, pp. 2270-2278.
- Aicha, M., Malika, B., Habiba, D. (2010) Two hybrid ant algorithms for the general T-colouring problem, *International Journal of Bio-Inspired Computation*, Vol. 2, No. 5, pp. 353 - 362
- Abdelaziz, A. Y. , Osama, R. A. and Elkhodary, S. M. (2012) Application of Ant Colony Optimization and Harmony Search Algorithms to Reconfiguration of Radial Distribution Networks with Distributed Generations, *Journal of Bioinformatics and Intelligent Control*, Vol. 1, No. 1, pp. 86-94
- H. Ahangarikiasari, M. R. Saraji, and M. Torabi, Investigation of Code Complexity of an Innovative Algorithm Based on ACO in Weighted Graph Traversing and Compare it to Traditional ACO and Bellman-Ford, *Journal of Bioinformatics and Intelligent Control*, Vol. 2, No. 1, pp. 73-78, 2013
- Dorigo, M. and Di Caro, G. (1999) The Ant Colony Optimization Meta-Heuristic, in *New Ideas in Optimization.*: McGraw-Hill, pp. 11-32.
- Michlmayr, E. (2006) Ant Algorithms for Search in Unstructured Peer-to-Peer Networks, in *ICDEW '06 Proceedings of the 22nd International Conference on Data Engineering Workshops*, Vienna, Austria
- Krynicki, K., Jaen, J., Mocholi, J.A. (2013) On the performance of ACO-based methods in p2p resource discovery. *Applied Soft Computing* Vol. 1, No. 12, pp. 4813-4831
- Association for Computing Machinery. (1998) ACM Computing Classification System (ACM CSS).
- Michlmayr, E. (2006) Ant Algorithms for Self-Organization in Social Networks. Vienna, Austria: Vienna University of Technology, PhD thesis.

List of Tables

Parameter	Interpretation	Value (RC-ACS)	Value (Semant)
TTL	Time to Live	25	25
q_0	Weight of exploiting vs. exploring strategy	0.80	0.85
R_{max}	Maximum number of resources to fetch	10	10
R_{min}	Minimum number of resources to fetch	5	5
α	Weight of newly deposited pheromone	0.07	N/A
w_d	Weight of resource quantity vs. link costs	N/A	0.5
β	Weight of link costs	1	1
γ	Factor in pheromone evaporation	0.02	N/A
ρ	Weight of evaporation	0.10	0.07
ph_{min}	Minimum pheromone level	0.001	0.001
ph_{max}	Maximum pheromone level	1	10000
ph_{init}	Initial pheromone level	0.009	0.009

Table 1 Execution parameters

Dynamism	Experiments	Expected impact	Example of a real-life analogue
Expansion	NetGrow 10k	Positive	Users create direct links to favourite sources
Contraction	NetShrink 10k	Negative	Communication distortion due to interference
Enrichment	ResGrow 10k	Positive	Nodes are equipped with new devices, users share more files
Reconfiguration	NodeMigrate 10k	Negative	Mobile users travel to distant locations

Table 2 Comparison of dynamism types

Experiment	Graph size	Growth Size	Initial Edge Set Size	Final Edge Set Size	Initial Average Vertex Degree	Final Average Vertex Degree
	$ V $	Δ	$ E _{init}$	$ E _{fin}$	$\overline{deg(v)_{init}}$	$\overline{deg(v)_{fin}}$
NetGrow10k 10links	1024	10	2560	2650	5.00	5.18
NetGrow10k 30links	1024	30	2560	2830	5.00	5.53
NetGrow10k 60links	1024	60	2560	3100	5.00	6.05
NetGrow10k 120links	1024	120	2560	3640	5.00	7.11
NetGrow10k 240links	1024	240	2560	4720	5.00	9.22
NetGrow10k 480links	1024	480	2560	6880	5.00	13.44

Table 3 NetGrow 10k experiments

Experiment	Graph size	Growth Size	Initial Resource Amount	Final Resource Amount	Initial Average Resource Amount	Final Average Resource Amount
	$ V $	Δ	$ Res _{init}$	$ Res _{fin}$		
ResGrow10k 10res	1024	10	30720	30810	30	30.09
ResGrow10k 30res	1024	30	30720	30990	30	30.26
ResGrow10k 60res	1024	60	30720	31260	30	30.53
ResGrow10k 120res	1024	120	30720	31800	30	31.05
ResGrow10k 240res	1024	240	30720	32880	30	32.11
ResGrow10k 480res	1024	480	30720	35040	30	34.22

Table 4 ResGrow10k experiments

Experiment	Graph size	Migration Size	Total Migrations	Migrations per node
	$ V $	Δ		
NodeMigrate 10node	1024	5% (51 nodes)	459	0.45
NodeMigrate 30node	1024	10% (102 nodes)	918	0.90
NodeMigrate 60node	1024	20% (204 nodes)	1836	1.80
NodeMigrate 120node	1024	50% (512 nodes)	4608	4.50
NodeMigrate 240node	1024	70% (716 nodes)	6444	6.30
NodeMigrate 480node	1024	90% (921 nodes)	8289	8.10

Table 5 NodeMigrate10k experiments

Experiment	Graph size	Shrink Size	Initial Edge Set Size	Final Edge Set Size	Initial Average Vertex Degree	Final Average Vertex Degree
	$ V $	Δ	$ E _{init}$	$ E _{fin}$	$\overline{deg(v)}_{init}$	$\overline{deg(v)}_{fin}$
NetShrink10k 10links	1024	10	2560	2470	5.00	4.82
NetShrink10k 30links	1024	30	2560	2290	5.00	4.47
NetShrink10k 60links	1024	60	2560	2020	5.00	3.95
NetShrink10k 120links	1024	120	2560	1480	5.00	2.89
NetShrink10k 240links	1024	240	2560	400	5.00	0.78
NetShrink10k 480links	1024	480	2560	0	5.00	0.00

Table 6 NetShrink10k experiments

List of Figures

Algorithm Start

1: Request R is generated by the vertex v_0 , it is routed by ant A . Take $i = 0$.

2: **loop** (i level)

2.1: Ant A attempts to satisfy R in the current vertex v_i

2.2: Choose randomly between exploration and exploitation

2.3: **if** exploitation **then**

2.4: Go to vertex

$$v_{i+1} = \operatorname{argmax}_{u \in V} \{ |\tau_\rho(v_i u)| \times |\eta(v_i u)|^\beta \},$$

where:

- $\tau_\rho(v_i u) \in (0,1)$ is the pheromone value on the edge $v_i u$, for the routing concept ρ
($\tau_\rho \in (0,1)$, if $v_i u \in E$; $\tau_\rho = 0$, if $v_i u \notin E$)
- $\eta(v_i u) \in (0,1)$ is the cost value on the edge $v_i u$
- $\beta \in (0,1)$ is a parameter

2.5: **else if** exploration **then**

2.6: Evaluate

$$p(v_i, u) = \frac{|\tau_\rho(v_i u)| \times |\eta(v_i u)|^\beta}{\sum_{u \in V} |\tau_\rho(v_i u)| \times |\eta(v_i u)|^\beta}, \text{ if } v_i u \in E$$

(*RC-ACS) Go to vertex v_{i+1} , chosen from all $u \in V$ with probability $p(v_i, u)$

(*SemAnt) Send a clone of the original ant A to every vertex u , each with probability $p(v_i, u)$

2.7: **end if**

2.8: Perform local pheromone update between v_i and v_{i+1} .

$$\tau_\rho(v_i v_{i+1}) \leftarrow (1 - \rho) \cdot \tau_\rho(v_i v_{i+1}) + \rho \cdot \gamma \cdot \max_{u \in V} \tau_\rho(v_i u),$$

where:

- $\rho \in (0,1)$ and $\gamma \in (0,1)$ are parameters

3: **until** R is not satisfied or A has not terminated

4: Perform global pheromone update for every edge vu in the solution S to the request R

$$(*RC-ACS) \tau_\rho(vu) \leftarrow (1 - \alpha) \cdot \tau_\rho(vu) + \alpha \cdot \frac{1}{|L|}$$

where:

- $|L|$ is the amount of edges in the solution S
- $\alpha \in (0,1)$ is a parameter

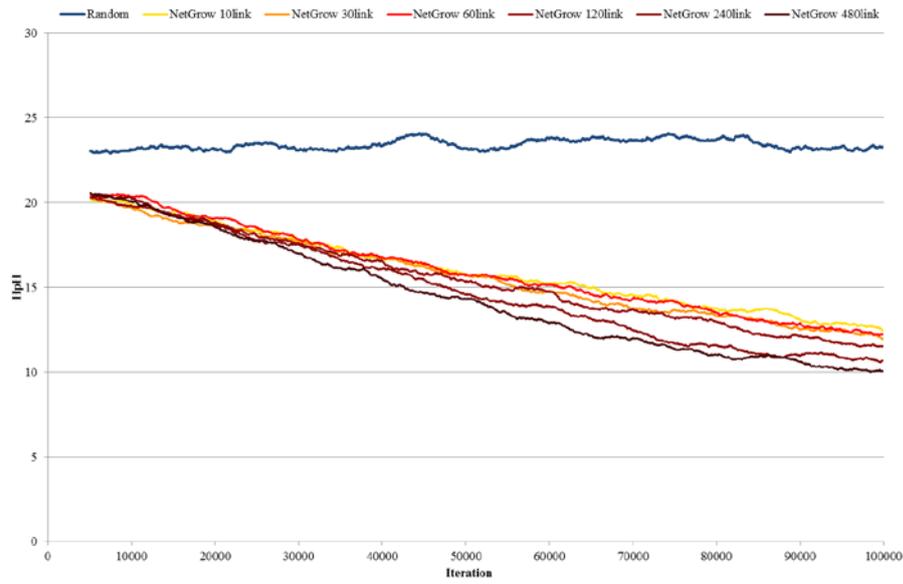
$$(*SemAnt) \tau_\rho(vu) \leftarrow \tau_\rho(vu) + w_d \cdot \frac{|S|}{S_{max}} + (1 - w_d) \cdot \frac{|L|}{2 \cdot L_{max}},$$

where:

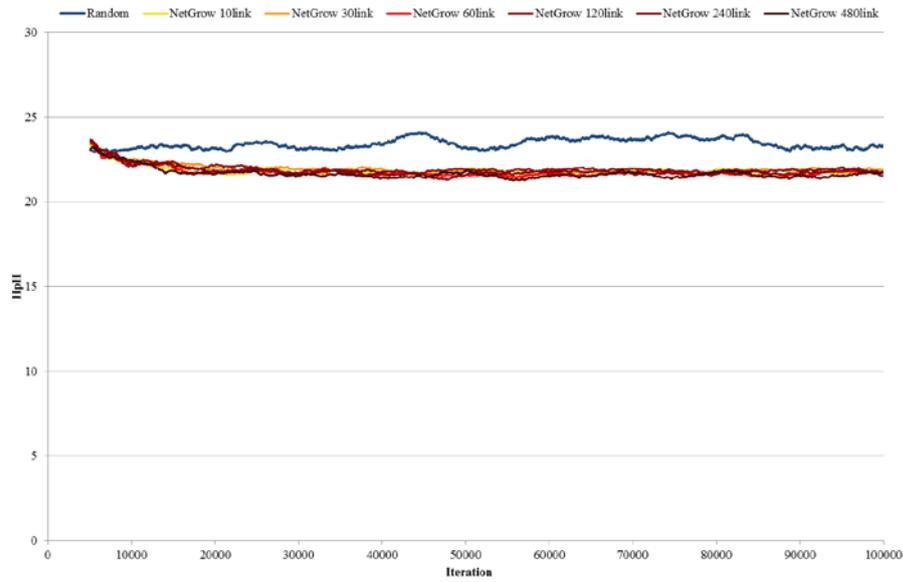
- $|S|$ is the amount of resources in the solution
- S_{max} is the maximum amount of resources permitted
- L_{max} is the maximum amount of edges in the solution permitted
- $w_d \in (0,1)$ is a parameter

End Algorithm

Figure 3-1 Algorithm 1: ACO-based query in graphs. ACS/SemAnt differences

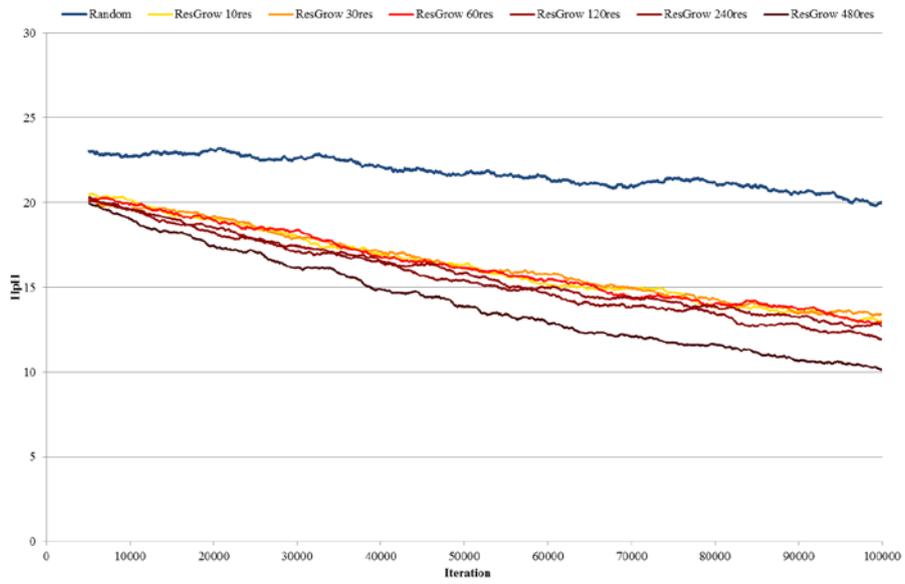


a) RC-ACS

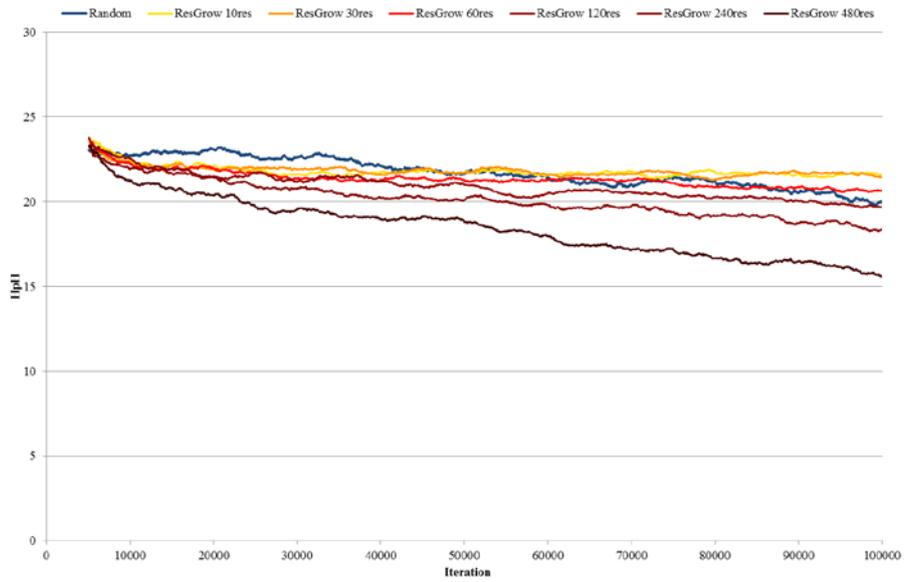


b) Semant

Figure 5-2 HpH in NetGrow10k: a) RC-ACS b) Semant

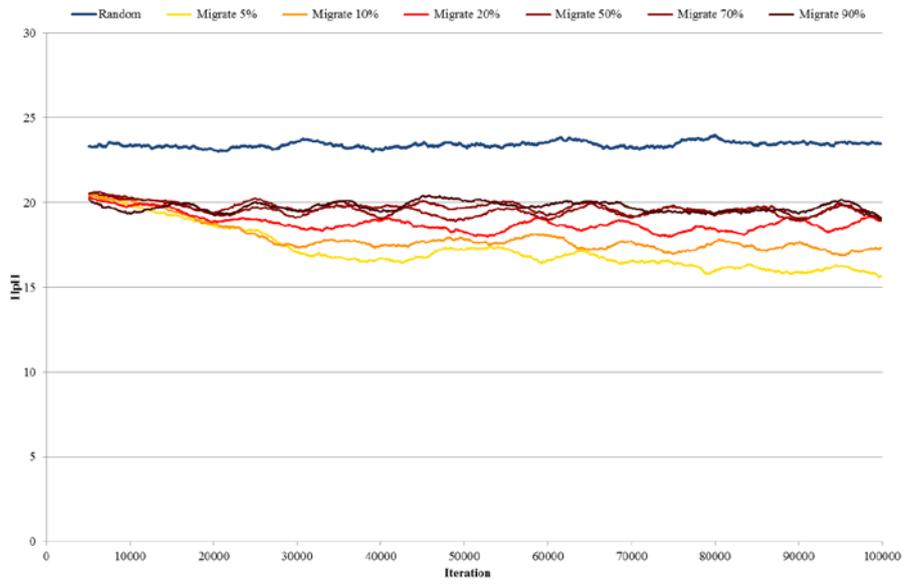


a) RC-ACS

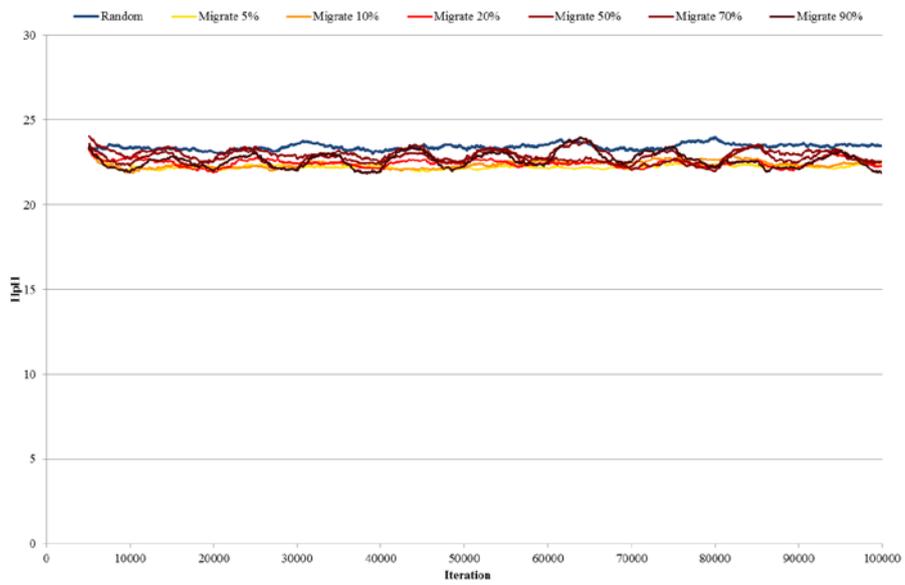


b) Semant

Figure 5-3 HpH in ResGrow10k: a) RC-ACS b) Semant

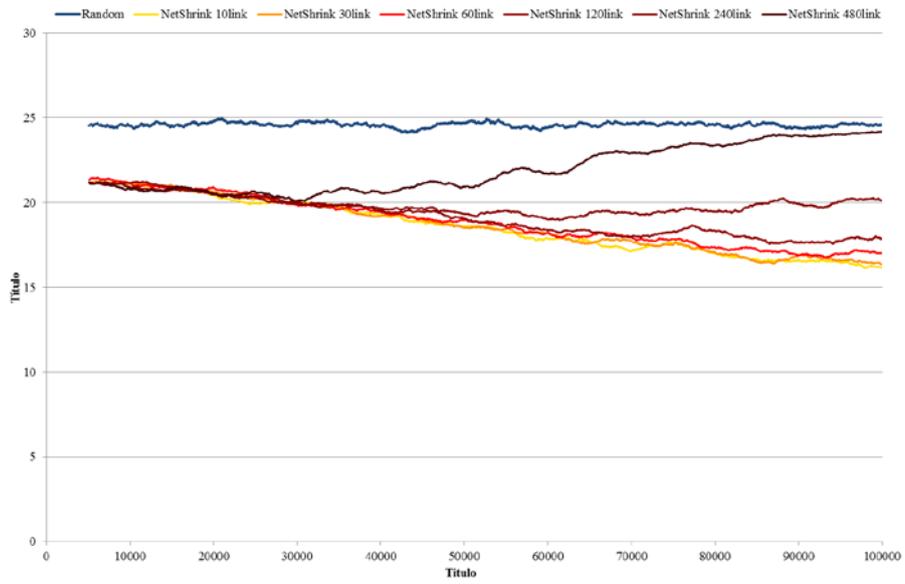


a) *RC-ACS*

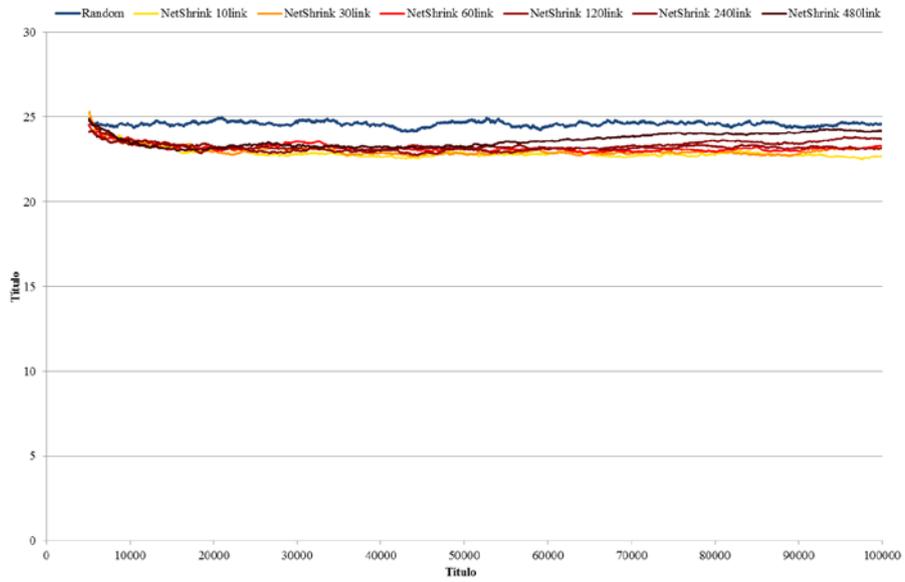


b) *Semant*

Figure 5-4 HpH in NodeMigrate10k: a) RC-ACS b) Semant



a) *RC-ACS*



b) *Semant*

Figure 5-5 HpH in NetShrink10k: a) RC-ACS b) Semant