

Web-agents inspired by ethology: a population of "ant"-like agents to help finding user-oriented information

Arnaud Revel

▶ **To cite this version:**

Arnaud Revel. Web-agents inspired by ethology: a population of "ant"-like agents to help finding user-oriented information. IEEE WIC'2003 : International Conference on Web Intelligence., 2003, Halifax, Canada. pp.482- 485, 10.1109/WI.2003.1241245 . hal-00426267

HAL Id: hal-00426267
https://hal.science/hal-00426267

Submitted on 2 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Web-agents inspired by ethology: a population of "ant"-like agents to help finding user-oriented information

Arnaud Revel
Equipe Traitement des Images et du Signal
ENSEA/UCP/CNRS UMR 8051
6, avenue du Ponceau,
95014 Cergy Cedex France
revel@ensea.fr

Abstract

In this article, we present a web-search ant-agent inspired by ethology and robotics. We detail its implementation on a set of FIPAOS platforms and show useful results in route finding and re-routing. Finally, we discuss its interest and drawbacks in comparison with classical search engines and give perspectives to overcome them.

1. Introduction

Finding information on the web is a tough task: the search space is so wide and the quantity of information is so huge that anybody would give up before the 100th click or the 100th line of text. Yet, paradoxically, the amount and density of available information could potentially lead everybody to find *the* peace of information he seeks.

A classical alternative is to use a search engine: the idea is then to create a centralized database and provide search tools to refine the search space. The main drawbacks of such a method are that: the information is centralized (and thus induces large resources – network, CPU, storage), the information extraction is mainly linked with statistical a priori and not with a feedback coming from users (even if search engines such as Google have integrated user feedbacks indirectly ranking the pages according to the number of hits to this page – [2]).

Facing these difficulties, we have tackled this problem keeping in mind our "animatician" background [8]. In this article we thus suggest a metaphor: let's suppose the web is a huge environment on which little software-robots could evolve. What would be their niche, their food and how would they evolve?

In the first part of this article we answer these questions. Then, we describe our system and the platform on which we develop it. Next, we present results and discuss them. Finally, we conclude and give perspectives.

2. Inspiration

Our research team is interested in designing neural control architectures for agents, either robotical or software. Besides, in order to guide the development of these architectures (but also to understand human and animal cognition) we are inspired by neurobiology and psychology and ethology.

In this framework, we have conceived a generic neural architecture called PerAc [6] to solve different robotical problems (in particular, we are interested in developing architectures which could allow a robot to solve different tasks during the same "life"). One of the application we develop concerns the navigation of a Koala^{TM} robot in a room or across rooms [3], which involves reinforcement learning or planning abilities.

Our claim is that a similar architecture could be used in a non-robotical context, turning the autonomous mobile robot into an autonomous mobile software-agent. Then, what is the equivalent of the environment in which the robot evolves and what is its goal and its niche?

In the context of Internet, the environment could be made of HTML/XML pages explored by the agent when it navigates on the web. Its goal would be then to find a precise information specified by the user. The niche of the agent would be the user's computer. Then, how to help the user finding interesting information? Which strategy using?

Our first approach has been to take inspiration from ethology, and in particular from Deneubourg's work [4, 5] on social insects. Indeed, Deneubourg has proposed models in which ants can spread chemical markers on their route to the food location (pheromones). Then, he has shown that several dynamical effects coming from the positive feed-

back due to pheromone deposit can emerge from the social interaction: route reinforcement based on social behavior and re-routing abilities. This model has been transposed to robotics (Gaussier94, unpublished), but also to other optimization problems, in particular by Dorigo [7]. Among those problems, Dorigo has tackled routing problems in telecommunication. Yet, to our knowledge, no application exists in information searching on the web. In the next section, we will present our own implementation of the concept of "web ants".

3. The "ant" web agents

We want to design software-agents acting as ants moving on the web, starting from the user's computer (niche) and looking for "food" (information). The main questions are: how to specify the searched information? How the "ant" agent could deposit "pheromone-like" markers?

In a first approach, we have simplified the problem considering the searched information is just a list of words the agent tries to match with the content of the target page (we will come back to this a priori in the perspectives).

Concerning pheromones, we have only access to intermediate http-servers on which pages are situated. The solution thus consists in depositing "pheromones" on the web-pages server. Thus, a *pheromone server* should exists in parallel with the http-server.

The role of the pheromone server is ambivalent: it must help the user to find information quickly, but it must also lead to the proper information. For that purpose, the route to the information corresponding to the user's desire (in our case, the page which matches it research pattern i.e. the list of words it searches) must be reinforced. When, the ant agent reaches a target page, all the pheromone counters along the path to the target page should be increased. Besides, in order to "forget" unused pathways, the pheromone level should decrease (reflecting the "evaporation" of pheromones).

The corresponding algorithm is detailed below.

Notations:

Let $Ph_k(t)$ be the pheromone level on a server s_k.

Let $\{s_1, ..., s_k, ...s_n\}$ be the set of servers directly reachable from s

While the current page does not content the list of searched words, the agent choose a page on server s_k at random with a probability equals to $p_k = \frac{ph_i(t)}{\sum_{l=1}^{k} Ph_l(t)}$ (see figure 1).

The pheromone quantity on each server s_k is updated according to the formula $\frac{\partial Ph_k(t)}{\partial t} = -\alpha \cdot Ph_k(t)$. If the current page contents the list of searched words, the agent comes back to its nest incrementing the pheromone servers according to formula $\frac{\partial Ph_k(t)}{\partial t} = \beta \cdot R(t)$, with $R(t)$ the reinforcement signal (see figure 2).

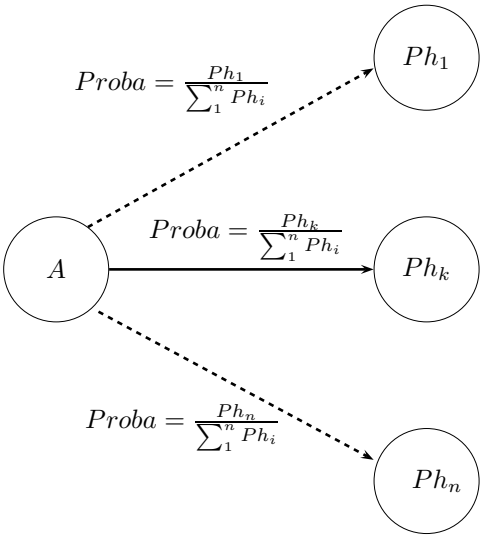


Figure 1. Choice strategy: the agent jumps on platform P_k (solid line) with a probability corresponding to the pheromone level Ph_k on this platform, normalized by the sum of pheromone levels of all the reachable platforms (dashed lines).

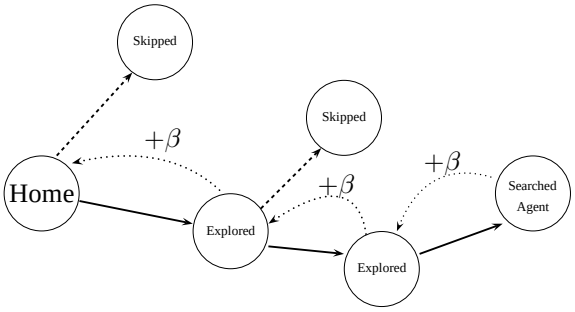


Figure 2. Reinforcement: when the information is found (solid lines) the agent jumps back home reinforcing the pheromone level of previous platforms (dotted lines). The unexploited routes are represented by dashed lines.

The next section presents our implementation of this algorithm on a FIPAOS set of platforms and gives practical precisions.

4. Implementation on a FIPAOS platform

This algorithm has been implemented on a set of FIPAOS platforms [1] in order to simulate the behavior of a population of agents on the web. We have simplified the HTML analysis phase considering the searched information was materialized by a specific "searched-agent" running on the target platform. During the search phase, the searching-agents have thus only to test if the searched-agent is on the same platform with no need to parse any HTML document. The purpose of the simulation is only to test the mobility of the searching-agents and their capacity to find the searched-agent faster.

Let A be the searching agent. Let S be the searched. Let Ph_k the level of pheromone on platform P_k. Let $succ(P_k)$, be the list of reachable platforms from P_k. Let $prev(P_k)$ be the list of previous platforms visited before P_k (in the reverse visit order – stack structure). Let $noise$ be a small random value (typically $noise \in [0, 0.001]$ – added to avoid division by zero).

"Ant"-agent A behavior

While S not found

  If A and S are on platform P_i

    Increase Ph_i

    Go to the nest incrementing $prev(P_i)$

  Else

    Decrease Ph_j ($P_j \in succ(P_j)$)

    Go to platform P_j with probability

    $\frac{Ph_j(t)}{\sum_{l=1}^{k} Ph_l(t)} + noise$

  End If

End While

Pheromone Ph_k updating

Increase

$$Ph_k(t+1) = (1-\alpha) \cdot Ph_k(t)$$

Decrease

$$Ph_k(t+1) = ph_k(t) + \beta$$

5. Results

In a first series of tests, we have tried to validate the convergence of the algorithm. For that purpose, we have used a fix network topology consisting of 3 platforms: the home-platform (P_H), the target platform (P_T) and an intermediate platform (P_I) (see figure 3). The home-platform was linked with both target and intermediate platforms. Pheromone levels were initialized to 0. The ant-agent was initialized on P_H and should find the searched agent on P_T using one of the two possible routes (direct or detour). We iterate this procedure 100 times. We test this setup with different value of α and β. As expected, we found: $Ph_{P_I} \to 0$ and $Ph_{P_T} \to \frac{\beta}{\alpha}$.

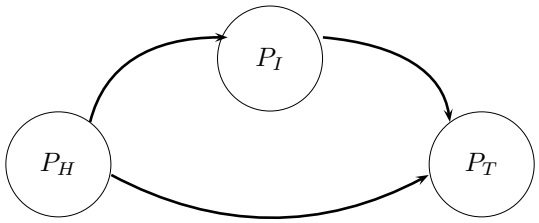


Figure 3. Tests network-topology. From P_H two links are possible to go to P_T: a direct link or a detour via P_I.

In a second phase, we wanted to test re-routing capabilities of such an architecture. Keeping the same setup, we have suppress the direct link between P_H and P_T. Obviously, the only other way being the one passing by P_I, the agent follows this route and the pheromone level converges to $Ph_{P_I} \to \frac{\beta}{\alpha}$. This setup being too simple, we have test a topology of 5 platforms with 2 indirect routes and found the same results: after cutting the shortest route, the agent use the second shortest route and the pheromone level still converges to $Ph_{P_I} \to \frac{\beta}{\alpha}$.

In order to test the interest of *mobile* agents we have also tested a search process performed remotely (the agent does not *move* but simulate the algorithm by communicating with remote platforms). Maybe surprisingly, even for simple topology, search-time is better using mobile agents rather than remote communication. Our interpretation is that, even if the agent moves, what induces the upload of the agent code on remote platforms, the number of communications is much smaller than in the case all communications are remote. In particular, the use of bandwidth is, for an important part, dedicated to synchronization and acknowledgment processes.

6. Discussion

In this section we want to stress the interests and drawbacks of the "ant"-agent approach comparatively with "classical" search-engine techniques.

In search-engine, the information about web-sites are centralized into huge databases. What is interesting is that

the search-engine location is well defined and can be accessed by everyone. Besides, it gives convenient tools to extract condensed information from raw data (database requests and pre-processing of HTML content). Yet, it has the drawbacks of its qualities: as the search-engine is centralized, many local resources (in term of CPU and bandwidth) are required (even if users requests are parallelized on different computers). Besides, what is more problematic is that there is no direct feedback of the user's satisfaction on HTML pages ranking.

In comparison, the use of "ant"-agents would permit to dispatch CPU and bandwidth use over the web according to each one own research (one searching for "cars" would probably not involve the same web-servers as one searching for "pets"). Besides, the user's satisfaction is directly used to modify the pheromone levels, and thus to reinforce routes that are "better", according to the user's interest.

The main drawbacks, of this approach is that specific *pheromone-servers* must be dispatched on the web in parallel with web-servers. Practically, it is difficult to assume every webmaster would accept to run a pheromone-server on is machine: the security risk is too important and simply *what would they have to gain?* In fact, we can propose an answer to this last argument, since if a webmaster provides "pheromone" abilities, maybe users would prefer using his site?

Moreover, the implementation we proposed is based on FIPAOS platforms which is a very resource-consuming java application. The reasons why both a webmaster and a user would not use our "ant"-agent is thus getting more important... An alternative would be to provide light-pheromone-servers and light-"ant"-agent, which could not properly use mobility but could remotely communicate with servers and use their properties.

7. Conclusion

In this article, we have proposed an "ant"-search-agent dedicated to user oriented information searching. We have shown its interest in route reinforcement, information searching refinement and re-routing capabilities.

In parallel with these interesting "theoretical" properties we have shown "practical" drawbacks linked with the difficulty of convincing both users and webmasters to use such agents. This is all the more dramatic that it has been shown that interesting dynamical effects could only emerge in rather large population of agents [4].

Our perspectives are thus to develop agent architectures internalizing representations of the best route to find the information searched by the user. Our previous experience in robotics should help us developing such "cognitive" agents [3].

In this article, we have also supposed that only one kind of pheromone exists. Yet, we refine could searching capabilities providing several pheromones which could be associated with a given kind of information or not. But, in this latter case, we are interested in demonstrating how a specific semantic could emerge from a coherent pheromone use by several users.

Finally, we have reduced information matching to a simple list words. Yet, this approach could be refined and, in particular, could be adapted to each user, by reinforcing user's criteria for satisfaction about the information it searches. It could considerably improve both information adequation and search speed. Besides, in a first phase, when no search about a given information has been performed, it could provide an a priori indication on the relevance of a given page and thus facilitate navigation. But how this process could interact with the route-reinforcement process?

8. Acknowledgment

This work is supported by the French minister of industry in the framework of the Netmarche project.

I would like to thank Sylvain Reynal for developing jpicedt (`jpicedt.sourceforge.net`) whereas there would be no figure in this article.

References

[1] http://fipa-os.sourceforge.net.

[2] http://www.google.com.

[3] A.Revel and P.Gaussier. *Biologically inspired robot behavior engineering*, chapter Designing neural control architectures for an autonomous robot using vision to solve complex learning tasks, pages 299–350. Springer-Verlag, 2003.

[4] J. 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 J.-A. Meyer and S. Wilson, editors, *From Animals to Animats: Proc. First Int. Conference on Simulation of Adaptive Behavior*, pages 356–363, Paris, France, 1990.

[5] J. Deneubourg, G. Theraulaz, and R. Beckers. Swarm-made architectures. In F. Varela and P. Bourgine, editors, *Proceedings of the first european conference on artificial intelligence*, pages 123–133. MIT Press, 1992.

[6] P. Gaussier and S. Zrehen. Perac: A neural architecture to control artificial animals. *Robotics and Autonomous System*, 16(2-4):291–320, December 1995.

[7] M.Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 1(26):29–41, 1996.

[8] J. Meyer and S. Wilson. From animals to animats. In M. Press, editor, *First International Conference on Simulation of Adaptive Behavior*. Bardford Books, 1991.