Ant Colony Optimisation for E-Learning: Observing the Emergence of Pedagogic Suggestions.

Yann Semet, Evelyne Lutton, Pierre Collet
INRIA – Projet FRACTALES
78150 Le Chesnay - France
Yann.Semet@inria.fr, Evelyne.Lutton@inria.fr, Pierre.Collet@inria.fr

***Abstract*— An attempt is made to apply Ant Colony Optimization (ACO) heuristics to an E-learning problem: the pedagogic material of an online teaching website for high school students is modelled as a navigation graph where nodes are exercises or lessons and arcs are hypertext links. The arcs' valuation, representing the pedagogic structure and conditioning the website's presentation, is gradually modified through the release and evaporation of virtual pheromones that reflect the successes and failures of students roaming around the graph.**

A compromise is expected to emerge between the pedagogic structure as originally dictated by professors, the collective experience of the whole pool of students and the particularities of each individual.

The purpose of this study conducted for Paraschool, the leading French e-learning company, is twofold: enhancing the website by making its presentation intelligently dynamic and providing the pedagogical team with a refined auditing tool that could help it identify the strengths and weaknesses of its pedagogic choices.

Keywords : E-Learning, Ant Colony Optimisation (ACO), Swarm Intelligence, Evolutionary Computation, Interactive Evolutionary Design.

I. Introduction

This work was initiated when Paraschool, the French leading e-learning company contacted the INRIA research center to conceive an automatic algorithm that would allow the relatively rigid albeit functional existing Paraschool software to behave differently depending on user specificities.

After several brainstorming sessions where neural networks, evolutionary algorithms and other artificially intelligent techniques were considered, it appeared that swarm-like algorithms could be used, thanks to the great number of actual users (more than 10000) and more especially ant-based probabilistic optimisation that could easily be grafted on the existing pedagogical graph constituted by the Paraschool software.

Moreover, Ant Colony systems present the interesting property of exhibiting emergent behaviour that allow individuals to benefit from the dynamic experience acquired by the collectivity, which means, in pedagogic terms that a student could benefit from the pedagogic lessons drawn out of his peers' successes and failures.

The implementation of these algorithms yields results that go beyond the requirements of the Paraschool company which will soon be experimenting in real size the automatic dynamic optimisation of the pedagogic graph (their set of interconnected lessons and exercises) implemented by their software.

This paper successively presents a concise description of human-learning concepts and their software implementation, a short description of the technical implementation of the Ant-Colony based optimisation algorithm and a discussion on the use of various selection operators. A set of experiments is then conducted, showing that erroneous arc probabilities can be automatically corrected by the system.

II. Elements on the Philosophy of Learning

The main concepts of teaching and learning used nowadays are still very old. The two main currents are Constructivism, that was elaborated by Kant and Behaviourism: a theory that came from Pavlov's experiments.

A. Constructivism

In 1781, Kant tried to synthesize rationalist and empiricist viewpoints. Kant sees the mind as an active agent, that organises and coordinates experiences. Along these lines, Piaget states that knowledge is not simply "acquired," by children bit by bit, but constructed into coherent, robust frameworks called "knowledge structures." Children are not passive absorbers of experience and information, but active theory builders.

Papert, a mathematician, and one of the early pioneers of Artificial Intelligence (he founded the Artificial Intelligence Laboratory at MIT), worked with Piaget at the University of Geneva from 1958-1963. This collaboration led Papert to consider using mathematics in the service of understanding how children learn and think [14].

In Constructivist theory [2], the emphasis is placed on the student: Teachers are seen as coaches who assist students to construct their own conceptualisations and solutions to problems. Collaborative groups with peer interaction are an important part of the learning process.

Such methods suppose that the student is given most of the elements allowing him to solve a problem (sometimes, the student needs to elaborate on his own missing elements), and he must use his intelligence to construct the solution. Such methods are not well suited to computer-based learning or e-learning software.

B. Behaviourism

Pavlov was studying reflexes in animals, and published in 1903 his famous experiment on how a conditioned salivation reflex could be created with a dog. His work was carried on by Thorndike, who came to the conclusion that learning is improved when it achieved a satisfactory result, with the extension that being wrong does not teach to correct errors.

Then, Watson published in 1913 "Psychology as the behaviourists views it," which, along with Pavlov's conditioning created a new paradigm. Those concepts were refined by Skinner in 1936 when he came up with the *Skinner Box* (to record animal or baby behaviour when confronted with levers, keys or discs that provided reinforces such as food and water) in which his second daughter spent much of her babyhood.

Skinner then developed teaching machines so students could learn bit by bit, uncovering answers for immediate "reward." These techniques were the source of the 1970/80 *drills and practice* (i.e. a series of exercises following the same pattern) and Multiple Choice Questions. Most computer-based self-instruction software are still based on Skinner's techniques, using generally a linear path, or branches for evolved systems.

The following stage is the *tutorial*, that appeared in the 70's. The main addition is *remediation*, i.e. giving necessary information to correct the error (when the answer is wrong) rather than a buzzing sound or the image of an exploding bomb on the screen. This means that the software should try to anticipate on possible answers, to be able to give the good remediation when the answer is wrong. Unfortunately, it is impossible for a deterministic software to anticipate all possible wrong answers, meaning that there is no real solution to this crucial problem with tutorials. Another problem with such software is that only one way is used to solve the problem among many possibilities.

Finally, the most advanced stage in behaviourism is Intelligent Tutorial Systems [11]. This technique combines tutorials with Artificial Intelligence techniques to analyse wrong answers and find the remediation that will be the most helpful to the student. Unfortunately, this means that the AI associated with the tutorial needs to be developed specifically for the domain addressed by the tutorial.

Current research is focussed on refining the AI algorithms, as well as collaborative learning environments, where students communicate over a wide or local area network [22].

III. Paraschool's E-learning Software and Requirements

A. Current Software

The internet-based e-learning software of the French Paraschool company offers a complement to high-school teaching in three quite different domains: Mathematics, Physics and French. The software is based on behaviourism and remediation (notion of tutorial), with short lessons, exercises and multiple choice questions. More than 10000 students are using the software, either individually at home, over the internet, or in groups in high-schools, over a local area network, with a supervising teacher. When working at home, if interaction with a human tutor is needed, the student can "chat" with a teacher on duty. Tougher questions are submitted and answered by mail within one day. The original software provides different subjects that often start with a short animated lesson (Flash file), followed by a series of exercises.

When entering an exercise, the system monitors the student activity, such as the time spent to submit an answer, so as to provide a global evaluation of the student on the given subject. If the answer is wrong, remediation is implemented under the form of corrective explanations that are given to the student. Parents or tutors may select subjects or exercises and include them into an agenda. Items do not disappear of the agenda until they are validated.

Although the system behaves differently depending on the answer provided by the student, the navigation remains deterministic in the sense that once a subject is chosen, all students will have the same list of exercises to solve, presented in an order that is determined by the pedagogical team. Individual or collective specificities (dynamic or static) are not taken into account.

B. Requirements

Paraschool was looking for a system that would enhance site navigation by making it adaptive to the user, according to the history of the system (recording of success and failure for each student, plus other parameters, such as the time spent on an exercise, etc.), so that both individual profiles and general time changing constraints could be taken into account.

Different students should see a different software, adapted to their level and specificities, as detected by the software or the tutor. Good students (who have a high rate of success) should not be bothered with elementary lessons which, on the contrary, must be submitted to students with a low success rate.

C. Proposed Solution

The numerous difficulties (multiple contradictory objectives, fuzziness, complexity) pertaining to this problem soon rang the bells of Soft Computing and Evolutionary Techniques, among which Ant Colony Optimisation (ACO) seemed particularly well suited.

Applying this technique to the Paraschool problem is relatively straightforward when the Paraschool application is seen as a graph with valuated arcs: nodes are pedagogical items, arcs are hypertext links and probabilities are associated to arcs. The numerous students are implemented as ants, that roam through the graph while leaving evaporating pheromones along the arcs they follow to reflect their rate of success.

Ant models and more generally Swarm Intelligence heuristics and their emergent properties seem to be especially adapted to this problem where one tries to come up with a solution that satisfies multiple and sometimes contradictory influences (from students, teachers and even exercises, which can be said to have an influence by being more or less difficult) in a gradual and dynamic fashion.

IV. Implementation of the Ant Colony: Algorithmic Overview

All nodes (html pages) of the new Paraschool software now contain a new ACO-powered $\boxed{\text{NEXT}}$ button that leads the user along an arc chosen by a selection algorithm (see section V), based on the probability associated with the arc. This probability is computed by taking several factors into account in the design of a weighted fitness function described in the next section. These factors are the following and play at both the individual and collective levels:

A. Pedagogic Weights: W

This *pedagogical weight* is the main value of each arc. It is implemented as a static (i.e. "global") variable (W), accessible to all ants. (W) is set by the Paraschool teachers and reflects the relative importance of the arcs that come out of a particular node. In other words, the teachers encourage the students to go toward such or such exercise after such or such lesson by giving the corresponding arc a higher weight. This valuation of the graph describes the *pedagogic structure* that will be optimized by the ACO algorithm

B. Pheromones: S and F

There are two kinds of pheromones that can be released on arcs to reflect students' activity :

S: success pheromone.
: This floating point value is incremented by ants/students on the adequate incoming arcs when they are successful in completing the corresponding exercise.

F: failure pheromone.
: This last value is S's counterpart for failure.

These pheromones are released not only on the arc that led the ant to that node but also on previous ones in the ant's history with decreasing amplitude. This is meant to reflect the fact that the outcome of a particular node (exercise) is influenced by all the nodes (lessons, exercises) the ant went through before but with an influence that, of course, diminishes with time. For obvious pragmatical reasons, this "back propagation" of pheromone release is limited in scope (a typical value of 4 has been agreed upon). To illustrate this, let us consider an ant that went through nodes A,B,C,D,E,F and that reaches node G. When it validates node G with success, 1 unit of success pheromone is dropped on arc (F,G), 1/2 unit on arc (E,F), 1/3 of a unit on arc (D,E) and 1/4 on arc (C,D).

In addition, to allow for dynamic adaptability of these pheromone amounts (S and F), evaporation is performed on a regular basis, usually every day, by reducing S and F in a given proportion τ typically around 0.999:

$$S_t = \tau * S_{t-1}; F_t = \tau * F_{t-1} \tag{1}$$

.

C. Personal History: H

This variable belongs to each ant, and bears information on visited nodes. Each time a node is validated, a history variable H, specific to each ant, is created, stored in database and set to $h_1 = 0.5$ if it is a success, to $h_2 = 0.75$ if it is a failure. This value will be later used as multiplicative factor to reduce the probability to visit that node again. When it eventually happens, H is again multiplied by h_1 or h_2. Just like S and F, H evaporates and tend to go back to 1 with time, following:

$$H_t = H_{t-1}\left(1 + \frac{1 - H_{t-1}}{H_{t-1}} \frac{1 - e^{-\tau x}}{1 + e^{-\tau x}}\right) \tag{2}$$

where τ is a constant used to tune the evaporation speed and x is the amount of time elapsed since the last visit to that node. τ should be calibrated to correspond to the volatility of the students' memory:

$$\tau = \frac{1}{x} ln\left(\frac{1+\alpha}{1-\alpha}\right) \tag{3}$$

with

$$\alpha = \frac{H_t - H_{t-1}}{1 - H_{t-1}} \tag{4}$$

Provided one defines what "forgetting an exercise" means, for instance if its H value, from $H_{t-1} = 0.5$ (one visit with success), grows back to $H_{t-1} = 0.9$, this gives $\alpha \approx 2.2$ and the pedagogic team then only has to estimate the time it takes to "forget an exercise": one week for example ($x = 604800 sec.$ gives $\tau \approx 3.6E - 6$

V. Selection Pprocedures: Control and Speed of the Decision Process

This section describes how an arc is selected when a user presses on the ACO-powered $\boxed{\text{NEXT}}$ button. The purpose of the ACO algorithm is to allow for an appropriate balance between the factors that should dictate an ideal pedagogic structuration: teachers' opinion, collective experience and individual history. For each arc a, this balance is represented by a "fitness" value:

$$f(a) = H(\omega_1 W + \omega_2 S - \omega_3 F)$$

The higher this value, the more "desirable" the corresponding arc will be and consequently the more likely will it be for it to be suggested to students. One sees that an arc is desirable when:

- its ending node was not visited or visited a long time ago (H close to 1)
- it is encouraged by professors (high W)
- there's an atmosphere of success around that arc (high S)
- little failure happened around that arc (low F)

One also can notice that the relative influences of the different factors can be tuned by tuning the ω_i values.

After a node has been validated, the outgoing arcs are sorted according to this computed fitness value. One arc is picked among the whole list following one of the selection procedures described below and is suggested to the student through the $\boxed{\text{NEXT}}$ button.

Five selection mechanisms are implemented:

Roulette-Wheel selection:

With this very traditional procedure ([8]), the probability for arc a_i to be picked is proportional to its fitness:

$$p(a_i) = \frac{f(a_i)}{\sum_{j \neq i} f(a_j)}$$

This method has the advantage to be entirely automatic and very sensitive to fitness variation. On the other hand, if an arc becomes really preponderent the others have really few chances to be selected. Besides, one has no control over the selection pressure.

Ranking based selection:

In this case, probability is reversely proportional to the arc's rank. The selection process is again entirely automatic, but it cancels the undesirable effects of large discrepancies in fitness values. As a side effect, subtle differences introduced by gradual release of pheromones might not be taken into account.

Ranking selection with manual thresholds:

This refinement of the previous method allows the designer to manually set the probabilities for an arc to be selected corresponding to its rank. This method gives complete control but necessitates heavy and probably non trivial parameterisation.

Tournament selection:

s_1 outgoing arcs are randomly chosen out of the total number of outgoing arcs and the best one is selected. This method is efficient and gives control over selection pressure with a single parameter.

Stochastic Tournament selection:

The worst arc is selected by default. $s1$ challengers are going to tried in a row. If the challenger's fitness is higher, it replaces the currently selected arc with probability s_2. Using 2 parameters, this method gives a more refined control over the selection pressure.

The pros and cons of the various selection procedures are quite well known in the context of Genetic Algorithms (see for example [9]). Although most pros and cons remain the same it is important to understand that the choice of these selections scheme is extremely important in the presented system.

The choice of the selection scheme and pressure s conditions:

Reactivity

s conditions the *speed* with which an arc, which emerges as excellent, reaches the state where it is systematically suggested to students. In other words, the higher the selection pressure, the faster an successful path is going to take over its neighbours. Controlling the velocity of such system is really a fundamental issue: a too reactive system would be chaotic and wouldn't make any sense while a too reluctant system would not reflect any of the dynamic the information brought by students.

Control

This system is interactive and as such, needs particular care regarding the way it is parameterizable. Many parameters allow for more freedom but require more efforts on time spent on calibration. The range of implemented selection procedures goes from entirely automatic to completely tunable with the possible intermediate steps of 1 or 2 parameters.

Signal

In this algorithm, arcs' fitness is a refined quantity constantly and smoothly updated by tiny stigmergic information. Choosing rank-based methods might lead to great information loss, unless large fitness discrepancies are observed.

The selection procedure must therefore be carefully chosen and tuned to get the appropriate amount of speed, control and information. As a compromise, most the presented experiments were realised with a stochastic tournament for the reason that it is reactive, tunable and safe regarding a not yet properly calibrated fitness function.

VI. Conducted Experiments

The described system has not yet been tried online with the 10000 students of Paraschool. It is therefore necessary to make sure as much as possible that expected behaviour occur before large scale real-world tests are conducted.

A. Simulation Hypotheses

A model of user population has been derived to conduct simulation tests:

Each ant, representing a virtual student, is given a certain *level*, represented by a floating point number situated between 0.0 and 1.0. This value is normally distributed over the population of students with mean 0.5 and standard deviation $1/3$.

Each exercise is assigned a *difficulty* value, also between 0.0 and 1.0. When an ant arrives at a given node, if its level allows it to validate the node (*level* $>$ *difficulty*), it succeeds, otherwise, it fails. Pheromones are released accordingly.

General calibration of the algorithm is performed on a "real" graph, i.e. corresponding to an actual part of the Paraschool website (the "Vectors" chapter of a mathematics course for high school students around age 14). Arcs between nodes and corresponding weights have been assigned by the Paraschool pedagogical team.

The sample case is therefore realistic (20 nodes, 47 arcs) and constitutes a meaningful structure in real size.

B. An Elementary Test Case

Several features are expected from the ant colony. In particular, it should be able to correct inappropriate arc pedagogical weight values.

To investigate this properly, after a rough calibration and observation process conducted on the real sized graph mentioned above, experiments are made on a reduced graph that exhibits such a situation, as illustrated by figure 1. After

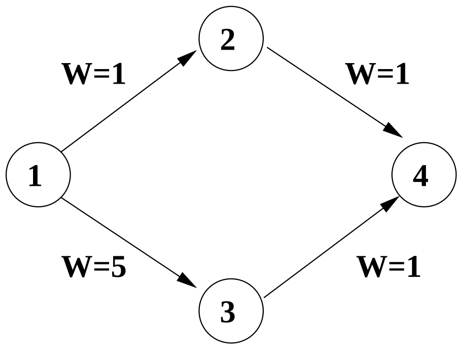

Fig. 1. The elementary test case is a situation that could be seen as inappropriate: Arc 1-3 is favoured by the pedagogical team, whereas the success rate for node 4 is much higher when the ants come from node 2 than when they come from node 3.

solving exercice 1, the student can either go to exercise 2 or exercise 3. Exercise 3 is encouraged by the pedagogical team as the arc leading to it is assigned a weight of 5 versus 1 for the arc leading to exercice 2, which however yields a five times greater probability for students to be sent to exercise 3 after exercise 1 is validated.

The problem is that the success rate of exercice 4 is much higher when the student comes from exercise 2 than when he/she comes from exercise 3. What is expected from the system in such a case is to detect the situation and to inverse the two probabilities so that students are encouraged to follow the right path. This should be achieved naturally, i.e. without any human intervention, thanks to the release of virtual pheromones along the arcs.

The arc leading to exercise 2 are going to hold a large amount of success pheromones and a low amount of failure pheromones. The arc leading to exercise 3, on the contrary is going to be in the opposite situation and this double discrepancy is going to be reflected in the arcs' fitnesses, thereby modifying their probabilities to be followed. Progressively, the arc leading to exercise 2 is going to take over the arc leading to exercise 3 and a reasonable situation should be promptly reestablished.

The next section describes experiments that were conducted to make sure that the system was able to behave as expected, and to determine the corresponding appropriate range of parameters.

VII. First Results

This section reports examples of numerical results obtained after calibration experiments conducted on the reduced graph.

A. General Calibration for Stability

The first lesson drawn from preliminary experiments is that without careful calibration, the system can behave in a variety of undesirable ways that includes fitness values quickly diving to zero (too strong evaporation), randomly growing to infinity (too strong α_s) or oscillating around their original values. The latter case is illustrated by the example shown here where the

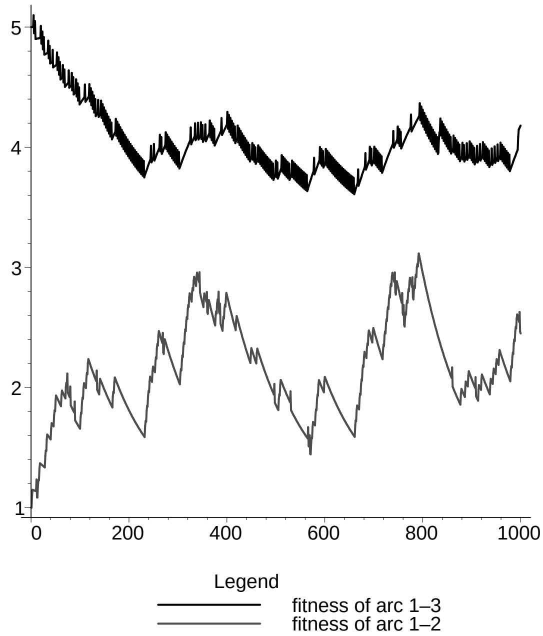

Fig. 2. Fitnesses of arcs (1,3) and (1,2) versus the number of iterations. $\tau = 0.99$ Oscillating behaviour, T^* does not exist. ($\alpha_s = \alpha_f = \alpha = 0.1$, $\omega_1 = \omega_2 = 1$, $\omega_3 = -3$)

evaporation rate τ is calibrated. Both curves (see figures 2 and 3) show the respective fitnesses of arcs (1,3) and (1,2) plotted versus the number of iterations, i.e. the number of times an ant goes from one node to another (when an ant reaches the end of the graph, a new one is placed at the starting node).

In the first case (figure 2), with $\tau = 0.99$, evaporation is too quick and the arc fitnesses don't have time to get inverted as they should.

Their values keep oscillating pulled up by pheromones and pulled back down by evaporation. With a higher $\tau = 0.999$, the second curve (figure 3) shows the appropriate expected behaviour: fitnesses are quickly inverted (at an iteration we will name T^* and remain inverted, either stabilising (f(1,3)) or growing at a reasonable rate (f(1,2)).

In this particular test case, the system is considered to be stable when it behaves this way. It means that T^*, the inversion iteration, exists and that the fitness values f(1,3) and f(1,2) remain properly inverted beyond T^*.

B. Refined Parameter Tuning

T^* appears to be a key parameter of the system, describing the velocity with which appropriate changes are made to the arc fitness distribution. The following set of experiments aims at understanding how T^* is influenced by the system's various parameters. This should allow for competent tuning of those parameters.

Curves shown in figures 4 and 5 plot the value of T^* averaged over several runs of 1001 iterations versus a range of values for the studied parameter. Two parameters, identified as key ones are investigated: α, the quantum of released pheromone (for these experiments, $\alpha = \alpha_s = \alpha_f$) and ω_3

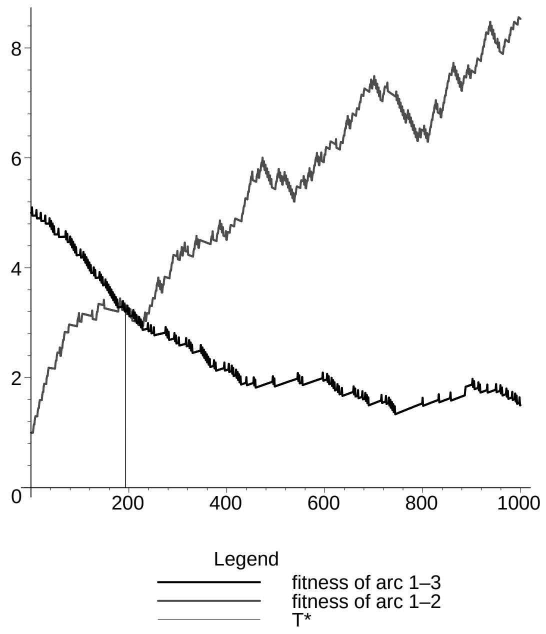

Fig. 3. Fitnesses of arcs (1,3) and (1,2) versus the number of iterations. $\tau = 0.999$ T^* exists, and is around 200. ($\alpha_s = \alpha_f = \alpha = 0.1$, $\omega_1 = \omega_2 = 1$, $\omega_3 = -3$)

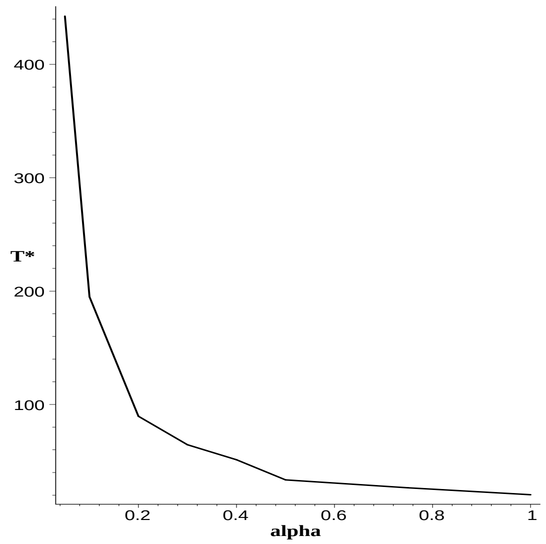

Fig. 4. T* versus α ($\tau = 0.999$, $\omega_1 = \omega_2 = 1$, $\omega_3 = -3$)

the weight corresponding to failing pheromones, which in the elementary test case carry the signal of the inappropriate situation as (1,3) holds a lot more failing pheromones than (1,2) due to the subsequent level of difficulty of node 4 (high if the ant comes from 3, low if it comes from 3).

Both curves show an exponential decay of T^* with respect to α or ω_3. A correct parameter setting can then be made based on the knowledge brought by those curves. For instance, optimal values can be fixed in order to yield a reasonably fast T^* without encouraging the system's instability with too high magnitudes.

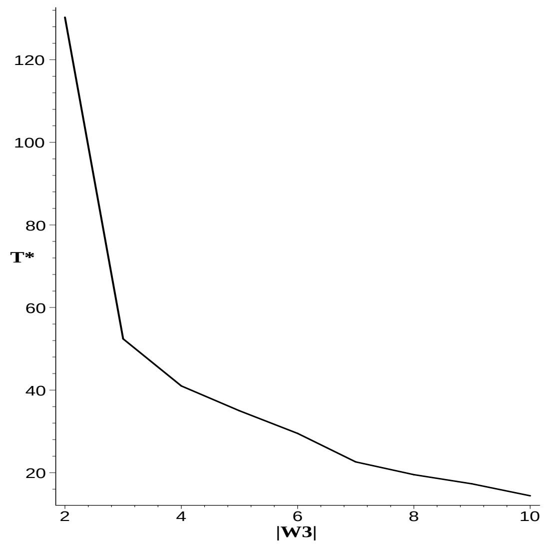

Fig. 5. T* versus ω_3 ($\tau = 0.999$, $\alpha_s = \alpha_f = \alpha = 0.1$, $\omega_1 = \omega_2 = 1$)

C. Real World Application

After simulation tests described above, the whole system has been connected to Paraschool's system and its entire pool of more than 10000 students. After a necessary debugging phase, it has been recently switched on in a silent mode where students' activity is only listened to (pheromones are recorded but arcs are not proposed yet). As the database is growing with recorded data, first observations seem to be encouraging as they allow to observe:

- Technical feasibility: the system works as predicted and without any noticeable overhead due to either system related or computational issues.
- How the navigation graph described by pheromone amounts structurates itself and gradually constructs its topology as students navigate through it.
- First hints on the weaknesses of the pedagogic structure as, for example,
  - singular nodes, i.e. too easy or too difficult exercises, are pointed at by large S/F discrepancies.
  - students' propensity to frequently get back to enjoyable items, such as those exhibiting funny pedagogic animations, is pointed out by overly charged arcs around those items.

Simulations tests and implementation preliminaries and first observations being encouraging, the system is about to be switched to active mode. The forthcoming and certainly crucial part of this work now lies in the observation of the interaction between students, professors and the system. These observations will be used for parameter calibration and feedback to enhance the system's design. It has to be underlined that this phase of observation and dialogue will be essential to the success of this project as it would be in any information system involving psychological interaction with human beings.

VIII. Conclusions and Perspective

Paraschool wanted a smart automatic system that could adapt to different users without manual intervention, which would be totally unrealistic to envisage on 10000 students.

The ant-based system described in this paper not only offers such automatic features by gradually modifying pedagogic paths suggested by teachers using collective experience and by making the structure individual-specific thanks to variables such as H but also comes up with emergent informations that can be used as a refined auditing tool to help the pedagogical team identify the strengths and weaknesses of the software and pedagogic material.

From a more theoretical standpoint, this work can be seen as a new take on Interactive Evolutionary Computation where the solution to a problem is gradually constructed and modified by multiple interacting entities with different and possibly opposite goals. A creative and robust compromise can be reached that balances all the influences and constraints, which allows all participating entities to benefit from an emergent culture and to enhance their decision making processes accordingly. This suggest a great deal of new and exciting applications in the field of Collective Cognition Modelling and Collective Evolutionary Design.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, *A volume in the Santa Fe institute in the Sciences of Complexity* Oxford University Press 1999. ISBN 0-19-513159-2.

[2] J. Bruner. *Constructivist Theory*. http://tip.psychology.org/bruner.html

[3] A. Colorni, M. Dorigo, V. Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life*, pages 134-142. Elsevier, 1992.

[4] J.L. Denebourg. The self-organizing exploratory pattern of the arhentine ant. *Journal of Insect Behavior*, 3:159-168, 1990.

[5] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD Thesis, Dipartimento di Electronica, Politecnico di Milano, IT, 1992.

[6] M. Dorigo, G. Di Carro and L.M. Gambardella. Ant Algorithms for Discrete Optimization. *Aritificial Life*, Vol. 5, No. 3, pp.137-172, 1999.

[7] D. E. Goldberg, The Race, the Hurdle, and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity, ILLIGAL Report No 98007, Urbana IL.

[8] D. A. Goldberg.*Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, January 1989.

[9] D. E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in *FOGA91*, vol 1, pp 69-93, 1991, also TCGA Report 90007.

[10] W. Jansen. The Added Value of E-learning. Informing Science. InSITE - "Where Parallels Intersect". June 2002. ecommerce.lebow.drexel.edu/eli/2002Proceedings/papers/Janse124Added.pdf

[11] D. Läge, S. Daub. The Resurrection of Intelligent Tutorial Systems: A New Way of Factual Knowledge Diagnosis. International Journal of Psychology, 35 issue 3/4, 136, 2000.

[12] D. Merkle and M. Middendorf, Modelling the Dynamics of Ant Colony Optimization Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, New York, 2002.

[13] M. Middendorf, F. Reischle, and H. Schmeck. Multi Colony Ant Algorithms. Journal of Heuristics (special issue on Parallel Metaheuristics), 8(3): 305-320, 2002.

[14] Seymour Papert. "Papert on Piaget." In *Time magazine's special issue on "The Century's Greatest Minds,"* p. 105, March 29, 1999.

[15] V. Ramos, F. Almeida. Artificial Ant Colones in Digital Image Habitants - A Mass Behaviour Effect Study on Pattern Recognition . In *Proc. of SIARP'2000 - 5th IberoAmerican Symposium on Pattern Recognition*, pp. 69-78, Lisbon, Portugal, 11-13 Sep. 2000.

[16] V.Ramos, J. J. Merelo, Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning. In *1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms, E. Alba, F. Herrera, J.J. Merelo et al. (Eds.)*, pp. 284-293, Spain, 6-8 Feb. 2002.

[17] M. Resnick, Turtles, Termites and Traffic Jams Explorations in Massively Parallel Microworlds. Complex Adaptive Systems series MIT Press, 1994

[18] D. Smith, G. Hardaker. E-Learning Innovation through the Implementation of an Internet Supported Learning Environment. In *Educational Technology & Society 3(3)* 2000. ifets.ieee.org/periodical/vol_3_2000/e04.pdf

[19] T. Stützle and M. Dorigo. ACO Algorithms for the Traveling Salesman Problem. In *Proceeding of EUROGEN conference M Makela, K Miettinen, P Neittaanmaki, J Periaux (Eds)*, John Wiley & Sons, 1999, ISBN: 0471999024.

[20] T. Stützle and H. Hoos. The Max-Min ant system and local search for the traveling salesman problem. In T. Baeck, Z. Mickalewicz and X. Yao, editors, *Proceedings of IEEE-ICEC-EPS'97 International Conference on Evolutionary Computation and Evolutionary Programming Conference*, pages 309-314. IEEE Press, 1997.

[21] T. Teich, M. Fischer, A. Vogel, J. Fischer. A new Ant Colony Algorithm for the Job Shop Scheduling Problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, California, July 7-11, 2001, p. 803

[22] T. L. Wentling, C. Waight, J. Gallaher, J. La Fleur, C. Wang, A. Kanfer, E-learning - A Review of Literature. Knowledge and Learning Systems Group, University of Illinois at Urbana-Champaign learning.ncsa.uiuc.edu/papers/elearnlit.pdf

[23] White T., Pagurek B and Oppacher F., ASGA: Improving the Ant System by Integration with Genetic Algorithms. In *Proceedings of the Symposium on Genetic Algorithms (SGA '98)*, July, 1998.

[24] D. Whitley. The GENITOR algorithm and selection pressure: Why rank–based allocation of reproductive trials is best. In Schaffer (1989), pages 116–121