



HAL
open science

An environment based approach for the ant colony convergence

Pierre Romet, Franck Gechter, El-Hassane Aglzim, Olivier Grunder,
Sidi-Mohammed Senouci

► **To cite this version:**

Pierre Romet, Franck Gechter, El-Hassane Aglzim, Olivier Grunder, Sidi-Mohammed Senouci. An environment based approach for the ant colony convergence. 11th International conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020), Nov 2020, Madeira, Portugal. hal-02879630

HAL Id: hal-02879630

<https://hal.science/hal-02879630>

Submitted on 12 Feb 2022

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.



Distributed under a Creative Commons Attribution 4.0 International License

The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks
(EUSPN 2020)
November 2-5, 2020, Madeira, Portugal

An environment based approach for the ant colony convergence

Pierre ROMET^a, Franck GECHTER^{b,c}, El-Hassane AGLZIM^d, Olivier GRUNDER^e,
Sidi-Mohammed SENOUCI^f

^aUniv. Bourgogne Franche-Comte, UTBM-CIAD, pierre.romet@utbm.fr, Belfort-90000, France

^bUniv. Bourgogne Franche-Comte, CIAD (UMR 7533), franck.gechter@utbm.fr, Belfort-90000, France

^cUniversité de Lorraine, Mosel Loria UMR CNRS 7503, franck.gechter@utbm.fr, Vandoeuvre-54500, France

^dUniv. Bourgogne Franche-Comte, ISAT-DRIVE Nevers, el-hassane.aglzim@u-bourgogne.fr, Nevers-58000, France

^eUniv. Bourgogne Franche-Comte, NIT-O2S, olivier.grunder@utbm.fr, Belfort-90000, France

^fUniv. Bourgogne Franche-Comte, ISAT-DRIVE Nevers, sidi-mohammed.senouci@u-bourgogne.fr, Nevers-58000, France

Abstract

Ant colony optimization (ACO) algorithms are a bio inspired solutions which have been very successful in combinatorial problem solving, also known as NP-hard problems, including transportation system optimization. As opposed to exact methods, which could give the best results of a tested problem, this meta-heuristics is based on the stochastic logic but not on theoretical mathematics demonstration (or only on certain well defined applications). According to this, the weak point of this meta-heuristics is his convergence, its termination condition. We can finds many different termination criteria in the scientific literature, yet most of them are costly in resources and unsuitable for practical problems. On the other hand, given the fact that the ACO is a stochastic approach, it seems difficult to decide whether to stop the algorithms in order to have the optimal result of the tested problems. Therefore, the thesis of this paper is to propose an approach based on the environment in order to determine the best termination criteria of the ACO, for an optimized solution.

© 2018 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Ant Colony, Environment approach, Dynamic convergence

1. Introduction

Ant colony-based algorithms were developed by Colorni, Dorigo and Maniezzo [1], [4], [5] in the early 1990's, and are now very common in the scientific literature. This meta-heuristics is based on adopting biological-inspired characteristics. It has then been applied to solve combinatorial optimization problems, called NP-Hard, and has been

*

implemented for the first time to solve the problem of the Travel Salesman Problem (TSP). Then, compared to other traditional optimization methods, it has rapidly widespread and implemented on problems such as task sequencing [2], or the color of graphs [3] benefiting from the use of the intrinsic properties. Besides, thanks to its feedback properties (positive and negative), to the preservation of its past experience in the form of pheromones and to its adaptation to the evolution of its model, this meta-heuristic is particularly suitable for dealing with combinatorial optimization dynamical problems. The determination of the termination criteria of ant colony algorithm remains in practice a sensitive research point. In this paper, we present our work focusing on this issue aiming at optimizing the number of iterations and characterizing the exploration properties of solutions when applied to a transportation system real case study. The emergence of the most appropriate terminating criteria for the situation is characterized by an analysis of the characteristics of the data model that we operate with. The paper is structured as follows: section 2 provides an overview of the basics of ant colony optimization algorithms. The section 3 is a non-exhaustive analysis of implemented methods to define the algorithm termination criteria. Then, the section 4.1 will be devoted to the analysis of ant colony convergence through local minimum exploration. Then section 4.2 is a environment based approach to characterize the ant colony convergence. The termination criteria being determined by the study of the environment. Finally, we will present you our experimentation, results and analysis into the 5 section.

2. Ant colony for operational research

This chapter is intended as a review of the main features of the ant colony optimization algorithm, that Dorigo synthesized in [7]. To do this, we will support our demonstration by the determination of the shortest path between two points; a combinatorial optimization problem already widely covered in the specific literature. For a set of "n" cities, a shorter path search allows to determine the smallest distance between two cities, passing through each of them once. The shorter path search is based on the implementation of a graph $G = (N, L)$, with N all cities and L all paths connecting cities. Each arc $li \in L$ having a d_{ij} value that characterizes the distance between two cities i and j .

Optimization by ant colony is the study of how work guides the worker [9]. Ant agents use pheromones that guide them on the paths with the shortest distance between the anthill and the food source. This solution is built by a succession of turns in which an agent travels guided by a pheromone trace and a search heuristic. Then, when all agents have finished their turn, they come back to the anthill by updating their pheromone trace. Ant agents thus seek a better solution within the paths that have the highest pheromone rate.

2.1. Iteration construction

When the algorithm is initialized, all ants (the number of ants being equal to the number of cities in the [6] graph) are dropped to the common starting point. Then, each ant agent applies a stochastic search, called the "random choice rule," to determine which to which city it will move on thereafter. Taking k agent ants, in a city i , the next city will be chosen with the probability defined below.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{r \in N_i^k} [\tau_{ir}(t)]^\alpha \cdot [\eta_{ir}(t)]^\beta} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

With $n_{ij} = \frac{1}{d_{ij}}$ a heuristic value, $T_{ij}(t)$ the value of the pheromone trace, with t the iteration counter. Side and Side parameters determine the influence of pheromone trace and heuristic value. Finally N_{ik} represents the direct neighborhood of the ant agent k , being in city i .

2.2. Pheromones update

Once all the ants have completed an exploration, they must return to their starting town and will at the same time update the pheromones rate of the paths forming their route during the last round. The pheromones are updated using the following equation:

$$\tau_{ij}(t+1) = [(1 - \rho) * \tau_{ij}(t) + mk = 1 \sum \Delta \tau_{ij}^k(t)] \quad (2)$$

With $p(0 < p \leq 1)$ evaporation rate, $\Delta\tau_{ij}^k$ the amount of pheromones that the agent k repents on the path visited, as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{f(S^k)} & \text{if the edge belong to } S^k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

here $f(S^k)$ is the size of the path constructed by the ant k agent, for the current iteration.

The pheromones act as a strengthening position, inducing other ants to use an already marked path, amplifying the positive reinforcement effect as more and more ants go along the path.

2.3. Termination condition

Since the development of optimization algorithms by ant colony, the efforts made through various research projects have focused on the development of iteration construction and pheromone updating. However, very few have sought to develop new approaches on how to end algorithms. In addition, it is legitimate to ask which methods to put in place, since the number of optimal solutions available is infinite.

3. State of the art of termination criteria

By the stochastic property of ant colony algorithms, the emergence of a solution for a given problem is guaranteed, with a probability of reaching 1, for an infinite time of exploration. Such application is not possible, therefore it is necessary to agree on the number of iterations to end ACO, to enable the emergence of an optimal result that best characterizes the exploited problem. The number of iterations being the key, the higher the iteration, the better the quality of the solution. However, arbitrarily setting a very high number of iterations is very inefficient in terms of resources, whether they are energy and/or computational. Therefore, it appears to be possible to use as little iteration as possible to determine the best solution. In addition, this search by means economy is adequate within this meta-heuristics, as illustrated by [15] through the "Stalling effect", which they define as the following: "the problem of stalling effect in fitness function is related to the non improvement of the fitness value during the evolutionary process". Since this phenomenon has been observed through many algorithm run, it appears that a better solution emerges at a given point in the research. However, the latter tends to stagnate before the simulation stops, without a better solution emerging again.

Thus, the optimization of the resources of ACO is entirely based on the way in which the termination criteria will be set. There are various methods in the scientific literature that we will now explain. Currently one of the most common approaches is to use an arbitrarily set termination criteria. This results in a high number of iterations, but above all, they are defined randomly. When the specified number of iterations is reached by the algorithm, the algorithm will return the best solution found [8]. Another, but similar, approach is to stop the iteration of the algorithm after it has exceeded a pre-defined execution time, always randomly [11]. In the end, these two approaches, although easy to put in place, reveal a complete decorrelation between the agents (ants) and their environment, cutting them off from the influence of the latter. In addition, they translate in a highly resource-intensive implementation, which is antinomic with the principle of optimization sought through meta-heuristics. Statistical approaches using evolutionary factors, are then used to evaluate the termination criteria of the algorithm to terminate its execution. Pascal Nicolas et al [12] have developed an iterative solution that involves launching their ant colony algorithm several times conditioned by a minimum number of iterations to reach. Then they determine a percentage of success by comparing the different solutions obtained according to the parameters of each execution. Zhaojun Zhang et al [16] propose a solution characterizing the evolution of the system using three factors. The first two assess whether the termination criteria is met and the last one is also an assessment factor based on the convergence of path with pheromone around the shortest path. In addition to the above methods based on the use of an arbitrarily fixed termination criteria, solutions based on decision trees were also developed. Kate A. et al [14] have developed a decision tree that determines the best heuristics to be implemented to solve the facing problem. Kate A. et al [13], have also worked on the ontology engine allowing to choose the best stop condition, allowing to optimize the termination criteria. Moreover, by exploring the literature to find advanced methodologies for our problem, a physics-centered solution was found. This solution

proposes to implement the mechanics of the point through the observation of position curves, speed and acceleration. The study of these curves is supposed to help in the decision-making process to stop the execution of the algorithm [15]. To finish this review of various approaches available in the literature, we will now discuss the search for the local minimum. The solution provided by [10], proposes to stop this search for a solution when a better solution is found, coupled with an arbitrary stop criteria in case the algorithm does not converge. However, nothing is defined as to how a better solution is evaluated, or even how it is correlated with the environment in which the research is conducted. Finally, the approach that we propose in this article is also based on this idea of local minimum exploration, like the solution of Silvia Mazzeo & Al [10]. However, unlike the latter, we strive to correlate the depth of exploration of the local minimum with the environment in which our agents operate. By doing so, we seek dynamic behavior for the algorithm, giving it an adaptation to its environment to help it make decisions about its termination criteria. By doing so, we are trying to demonstrate that a good knowledge of the studied environment can bring real added value in the search for solutions.

4. Different approaches to characterize the ant colony termination criteria.

4.1. Minimum local optimization

We have seen this since the beginning of this article and from the literature research that has been carried out, the definition of a suitable termination criteria remains a fundamental problem for optimization by ant colony. Indeed, more than stopping the program, the termination criteria implicitly defines the best possible time to stop the exploration of the problem. Without endless time for the best solution to emerge, it becomes necessary to be able to put in place a solution that could stop the execution of the colony when a better solution is achieved. Following the methods we saw in the previous part and through the "stalling effect", it seems appropriate to consider the determination of the local minimum, in order to determine whether it can optimize the stoppage of the search for solutions for the problem under consideration. As previously noted, Silvia Mazzeo [10] suggested an approach where until the solution found improves, the algorithm must continue to run. However, this method is limited by an arbitrary criteria, a minimum number of iterations, which must shunt the local minimum if it does not converge. Following the reading of this work and the inability of the decision-making and learning algorithms to determine the local optimum of a stochastic process, it was decided that we would base our research on the criteria of termination, through optimization of local minimum. As we have already pointed out, the ant colony for shorter-path search is a stochastic process, the determination of the most suitable solution is based on the study of the fitness value curve. This study makes it impossible to determine an optimal solution, without an infinite time of calculation. The only way we can exploit the latter is through real-time analysis of its evolution. The two algorithms we propose are based on the same idea, namely the analysis of the stability of the curve.

4.1.1. First Method:

Regarding the first method, we set a slippery average moving on the curve. At each time step, an average is calculated consisting of a set of values sampled on the curve. Then we calculate a delta, being the average subtraction: (AVERAGE-1)-AVERAGE. As this delta evolves at every step of algorithms, we compare it to the epsilon criteria, and when it becomes less than epsilon, we can stop our algorithm.

Algorithm 1 Moving average

```


$$\phi \leftarrow \frac{\sum_{i=1}^n K_i}{n}$$

if  $\phi < \epsilon$  then
   $\tau \leftarrow 1$ 
else
   $\tau \leftarrow 0$ 
end if

```

With ϕ the number of iteration, ϵ to the limit the stagnation of the quality solution and τ the criteria to stop the algorithm execution. However, this method reveals two shortcomings in its design. The first is that before we can calculate our average, we need to define a number of components of the average, and this is done stochastically. In addition, we compare our delta to a value that will also vary according to the person developing the solution.

4.1.2. Second method:

This second method is a refinement of the first, in that it allows us to eliminate a stochastic evaluation criteria. This is how we always have a criteria to define, but as we will see in the next part, the way we approach it should make it adaptive to the problem being explored. With respect to this second method, our algorithm is defined as follows. We initialize an Epsilon variable being the depth of the local minimum to explore, then for each possible case, we make this counter evolve. In the case of an improvement of the solution we increment Epsilon, then if it remains stable, we decrement it. Through this method, the higher epsilon is initialized, the deeper the local minimum is explored.

Algorithm 2 Calculate optimal number of iteration

```

 $\phi \leftarrow \text{reward}$ 
if  $\epsilon < \theta$  then
     $\text{reward} \leftarrow \text{reward} + 1$ 
     $\theta \leftarrow \epsilon$ 
end if
if  $\epsilon < \theta$  then
     $\text{reward} \leftarrow \text{reward} - 1$ 
     $\theta \leftarrow \epsilon$ 
end if
if  $\text{reward} == 0$  then
    Stop algorithm
end if

```

With ϕ the minimum number of iteration to limit the stagnation of the quality solution, ϵ the quality of the current solution, θ the quality solution of the previous iteration and τ the criteria to stop the algorithm execution. This first work on local minimum, although functional, is, however, "a problem" in its approach. As we have already pointed out, ant colony is a stochastic problem, a local minimum other than that obtained with a probability of one, is only an approximate solution. All of this to show that any outage of an ant colony search can only be valid if the outage is defined as optimal for the given problem. In other words, an algorithm, even if it is based on advanced programming methods such as machine learning, cannot be based on an arbitrary criteria to stop this search for a solution. To put such methods in place is to place at the heart of our method a significant chance factor. After that, it is always possible to weigh the meaning against the chance factor, because in the best case, any person setting up such an algorithm may have a sense as to how long his algorithm will last, or, more clearly, any person with sufficient knowledge of his field, may "predict" with more or less accuracy his termination criteria. However, in the rest of the cases, this criteria will be either well defined, but randomly, or in the worst case simply bad. Doing so is like playing with chance to bring about a better solution. Moreover, even if a great deal of optimization is done (and, moreover, in a beneficial way) on the realization of the ant towers, why rely on a stochastic method that can counterbalance the value gained from consuming resources and overburdening. We will therefore see in the rest of our work, whether it is possible to rely on the attributes of a problem studied to bring out an optimal local minimum and if so, what method(s) allowed us to define the characteristic attributes

4.2. Environment based knowledge

We discussed it in the previous part, the objective that we want to achieve through this research work is the characterization of the convergence of the ant colony algorithm. As we have seen, whether the solutions in the literature, or those we have proposed, they always end up using a criteria arbitrarily set to limit the solution. Based on this observation, we tried to take the counterpart of this design bias to think of an innovative methodology. Thus, if all arbitrary criteria are set aside, the question of how and on what basis the characterization of the emergence of a solution

should be raised very quickly. Since each project is different, the idea of an adaptive or dynamic solution is obviously becoming the way to go. From this observation, we must now find out what to base the dynamicity of our algorithm on. As said, every case studied through an ant colony is unique. The uniqueness of each graph can be reflected in its number of nodes, the number of arches or even complexity, as well as its orientation. So what we are going to do is to look at whether it is possible to highlight the intrinsic properties of the graph that can be correlated in order to define an optimal termination criteria, and so make any ant colony algorithms dynamic. To conduct this study, we have developed two test bases on which to test our research. The first one is a small graph, made of 10 vertices, but with a significant complexity for its dimensions. The second base is a reproduction of the public transportation system of the city of Belfort. This graph, although very voluminous, with 150 vertices, is not very complex. So it is on these graphs that we conducted experiments and generated raw data sets with them, and then we analyzed and refined raw data to make them speak. In the next section "Experimentation, Analysis & Results", we will see if one or more correlations allow the characterization of an optimal dynamic termination criteria for the ant colony algorithm. As seen before, from now on, articles found in literature, and which has been exposed above such as [?] are tending to follow the path of the exploration and the characterization of the stalling effect curve. However, some articles published in the early 2000's such as [6] followed another path exploring the influence of the number of ants as compared to the number or vertices in the graph. Indeed, Dorigo [6] estimate that with a number of ant equals to the number of vertices, the optimal number of iteration is reached. However, this solution does not take into account the intrinsic complexity of the graph. Results presented in section 5 show the importance of the complexity of the graph in the choice of the number of ants.

5. Experimentation, Results and Analysis

5.1. Experimental process and goals

According to the purpose of this paper, the main goal is to provide to ACO a full dynamic algorithm, by redefining and deleting old arbitrary criterion. Through the next two algorithm and based on the study of the stalling effect, we offer to develop this subject by a naive and simple way. As introduce earlier, some other more complex methodologies try to characterized the termination condition of the ACO. Despite, although functional those work always finish to used arbitrary criterion at one time or another. By the way, carrying on this way should allow us to simplify our approach, by reducing criterion to an absolute minimum and identify the purpose of the termination criterion. Furthermore, always in this idea of naive and simple way of thinking, the last implementation explore the idea of a self based determination approach. Having been able to reduce to a fundamental criteria, a environment based approach seems to be a promising lead to characterized the temrination criterion of the ACO.

Finally all our algorithms have been exploited on two different graphs. The first being manually generated and the second being a reproduction of the Belfort public transport system.

5.2. First method

We explained in the previous section, we have a slippery average so that we can stop the execution of our algorithm when the quality of the solution curve stagnates long enough. This algorithm is a first approach that can be considered naive. Although functional, it combines all the flaws of a previously cited solution, being based on several arbitrarily set criteria. The first arbitrary criteria being the size of the buffer on which we calculate our average. Then we compare our calculated average value to an arbitrarily fixed epsilon value.

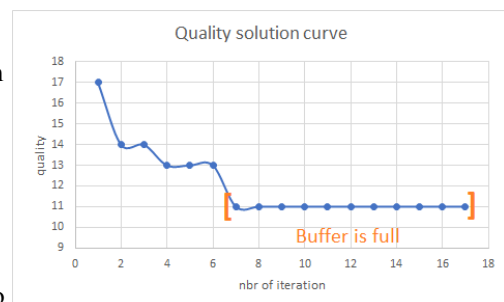


Fig. 1: First Method Results

5.3. Second Method

This second method allow to remove unnecessary arbitrary criteria to only keep the essential, just one. At the end, the objective is to determined this termination criteria thank to intrinsic properties of the graph explored. However, before attempting to dynamically defined this one, we will try to find if our algorithm has the desired behavior. As we can see on the image on the right, the algorithm evolves thanks to the quality solution found by the ants, improving or not at each iteration. As soon as a sufficiently long quality stagnation is detected by the algorithm, this last turn off.

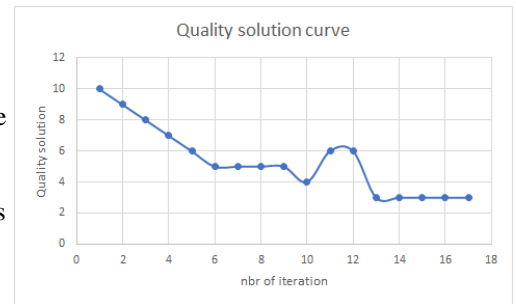
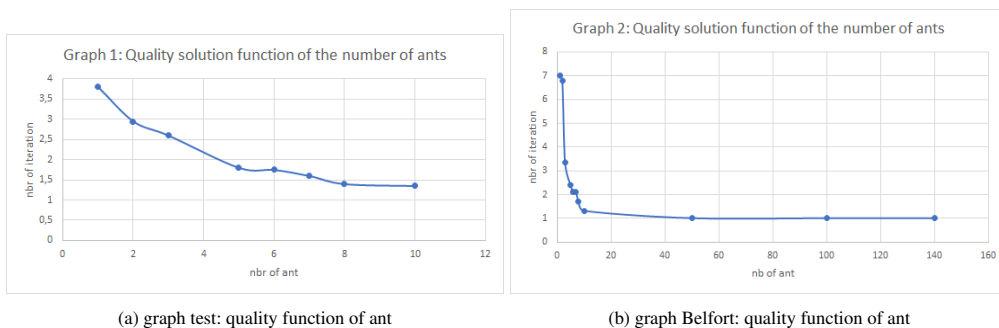


Fig. 2: Second Method Results

5.4. Ants number vs. iteration numbers

Having been able to demonstrate in the previous part that the behavior of our algorithm was well suited to our expectations, we will now try to determine the termination criteria of our algorithm based on the environment in which our ants operate. To do this, we started by trying to see if the number of vertices could influence the convergence of the ACO. So, we launched our ant colonies on our two graphs, one consisting of 8 vertices and the other of 150 (it is also important to note, at this stage of the analysis, that our number of ants is equal to the number of vertices, as explained in [6]). This operation has been repeated 100 times, in order to have an average number of iterations for convergence that is exploitable. However, this initial analysis did not achieve the expected result. Indeed, after completing our program, our two colonies converged with the same number of iterations, as seen under (a) (b), proving that the number of vertices did not affect the convergence time (in number of iterations) of the algorithm. We then took the part of questioning the results of Dorigo [6], this time varying the number of ants for each graph available to us. To do this experiment, we ran our ACO algorithm 20 times for every number of ants.



(a) graph test: quality function of ant

(b) graph Belfort: quality function of ant

It was during the latter experiment that conclusive results emerged. As we can see from the two graphs above, we were able to point out that the time of convergence (in iteration) of the algorithm depended indeed on the number of ants instantiated in the graph. However, we can see that the two curves do not have the same equation, which unfortunately does not validate our hypothesis, since the colony's convergence time was dependent on the number of ants.

6. Conclusion

Following a re-contextualization of the ant colony algorithm in the 2 section, we made available a non-exhaustive state of art regarding the COA termination criteria in the 3 section. We then introduced two new methods to characterize the convergence of the ant colony algorithm in the 4.1 section, based on local minimum exploration. Finally, we explored the path of an algorithm based on the intrinsic characteristics of the model explored. However, when we tried

to validate our hypotheses in the 5 section, the results that we obtained made us realize that we were not following the right path regarding the convergence of ant colonies. From our results, we were able to show that the number of Vertices had no influence on the time of convergence of the ACO and that, against all odds, the time of convergence depended on the number of ants instantiated in the graph. However, although the convergence time may be dependent on the number of ants, the study of the two graphs in section 5 shows that the "iteration convergence" function of the two graphs did not have the same equations. So we can conclude that our assumption is not validated, that the number of ants instantiated in a graph does not define the time of convergence of the algorithm. It therefore appears that another intrinsic feature of the graph should govern the time of convergence of the ACO. At present, the most likely track at our disposal would be that the complexity of the graph is the characteristic that interests us. As we can see, the only unexploited difference between our two graphs is their complexity. Using the two graphs section 5, it can be seen that, although the first is small, the latter requires a higher convergence time than the graph of the Belfort public transport system. The latter, which is smaller in size, is more complex, or in other words, has a larger number of connections per vertices. So, we are going to continue our research in this direction, to determine if the complexity of a graph is the key to automatic convergence of ant colonies algorithms.

7. Acknowledgements

This work is done with the support of the Région Bourgogne Franche Comté through the SURATRAM Project.

References

- [1] Alberto Colomi, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life*, 01 1991.
- [2] Alberto Colomi, Marco Dorigo, Vittorio Maniezzo, and Marco Trubian. Ant system for job-shop scheduling. *STATISTICS AND COMPUTER SCIENCE*, 34, 01 1994.
- [3] Costa D and Alain Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295–305, 03 1997.
- [4] M. DORIGO. Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*, 1992.
- [5] M. Dorigo, V. Maniezzo, and A. Colomi. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.
- [6] M. Dorigo, V. Maniezzo, and A. Colomi. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.
- [7] M. Dorigo and T. Stützle. Ant colony optimization algorithms for the traveling salesman problem. *Ant Colony Optimization*, pages 65–119, 2004.
- [8] Yuvraj Gajpal and Prakash Abad. An ant colony system (acs) for vehicle routing problem with simultaneous delivery and pickup. *Computers Operations Research*, 36(12):3215 – 3223, 2009. New developments on hub location.
- [9] Pierre-P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis* et *etcubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–80, 2005.
- [10] Silvia Mazzeo and Irene Loiseau. An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics*, 18:181 – 186, 2004.
- [11] Rizzoli A. E. Donati A. V. Montemanni R., Gambardella L. M. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10:327 – 343, 2005.
- [12] Pascal Nicolas, Frédéric Saubion, and Igor Stéphan. Answer set programming by ant colony optimization. In Sergio Flesca, Sergio Greco, Giovambattista Ianni, and Nicola Leone, editors, *Logics in Artificial Intelligence*, pages 481–492, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [13] Kate Smith-Miles, Jano van Hemert, and Xin Yu Lim. Understanding tsp difficulty by learning from evolved instances. In Christian Blum and Roberto Battiti, editors, *Learning and Intelligent Optimization*, pages 266–280, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [14] Kate A. Smith-Miles, Ross J. W. James, John W. Giffin, and Yiqing Tu. A knowledge discovery approach to understanding relationships between scheduling problem structure and heuristic performance. In Thomas Stützle, editor, *Learning and Intelligent Optimization*, pages 89–103, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [15] M. Veluscek, T. Kalganova, and P. Broomhead. Improving ant colony optimization performance through prediction of best termination condition. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2394–2402, 2015.
- [16] Zhaojun Zhang, Zuren Feng, and Zhigang Ren. Approximate termination condition analysis for ant colony optimization algorithm. In *2010 8th World Congress on Intelligent Control and Automation*, pages 3211–3215, 2010.