2015 IEEE Symposium Series on Computational Intelligence

Towards a Network Interpretation of Agent Interaction in Ant Colony Optimization

Pavel Krömer, Petr Gajdoš, Ivan Zelinka

Department of Computer Science
VŠB Technical University of Ostrava
Ostrava, Czech Republic
Email: {pavel.kromer, petr.gajdos, ivan.zelinka}@vsb.cz

***Abstract*—This work introduces a novel framework for a network interpretation of agent interaction in ant-inspired algorithms. A complex network interpretation of population dynamics is a recent trend in the research of population-based metaheuristic algorithms. Complex network models of nature-inspired methods enable the use of a wide variety of analytical methods from the areas of graph theory and network science in the field of computational intelligence. Agent interaction is in this approach represented by an evolving complex network with vertices corresponding to individual agents and arcs with changing weights quantifying their interaction. This paper presents a generic framework for such network interpretation of ant interaction as well as its initial implementation for a sample problem, the travelling salesman problem. Initial computational experiments illustrate the proposed concepts and demonstrate the usefulness of network-based analysis of ant-inspired methods.**

I. Introduction

It is well-known that ant-inspired algorithms operate on a network (graph) model of an underlying problem. The graph represents all possible problem solutions and, in fact, embodies a solution space that needs to be explored. In this space, various ant colony optimization (ACO) algorithms implement different flavours of a distributed cooperative path-finding process based on an indirect communication between agents (ants) via the deposition and evaporation of artificial pheromones. Multiple problem solutions are in such algorithms incrementally constructed using a sequential decision process [8].

Various graph properties of the solution space can be used to characterize different aspects of the problem and to obtain valuable suggestions regarding algorithm parameters. For example, in the $\mathcal{MAX} - \mathcal{MIN}$ ant system for the traveling salesman problem (TSP), the number of ants often depends on the number of vertices [28] and the lower and upper bound of the amount of pheromone can in the same algorithm, used for a water distribution design problem, depend on the average vertex degree [30].

The interpretation of the dynamic behaviour of evolutionary algorithms as a complex network of interactions between candidate solutions (population members) and the analysis of static and dynamic properties of such network is a recent research trend [6], [7], [31]. The investigation of the relations between attributes of such networks and the properties of the modelled algorithms opens a number of research questions and provides a new class of instruments for a more efficient control of nature-inspired metaheuristic methods [31]. Such studies were already presented for several variants of the differential evolution algorithm applied e.g. to the flowshop scheduling problem [6] and the permutative flowshop scheduling problem [7].

This work takes a similar approach and outlines a conceptual framework for a complex network representation of the interactions between ants in ant-inspired algorithms for permutation problems. In a permutation problem, the goal of each ant is to find a permutation of exactly n elements and the solution space is usually represented as a complete graph on n vertices. The dynamic representation of ant interactions in the form of a complex network introduces an auxiliary graph structure that is complementary to the graph defining the solution space. It allows analyzing the dynamic aspects of ant-inspired metaheuristics by established formal methods from the areas of graph theory, social network analysis, and e.g. network science.

The proposed network interpretation is illustrated on the travelling salesman problem. Although a simple problem with rather small instances that can be efficiently solved by a variety of other methods, it illustrates the proposed concepts very well and allows an initial assessment of the correlation between algorithm properties and selected attributes of the interaction network.

The remainder of this article is structured as follows: the fundamentals of ant-inspired algorithms are summarized in section II. The same section also provides a brief overview of several earlier formal approaches to ant-inspired algorithms. The proposed complex network interpretation of ant interactions in ACO for permutation problems is developed in detail in section III. Experimental evaluation of the proposed concepts on a real-world permutation problem, the TSP, is presented in section IV. Finally, major conclusions are drawn and future work is outlined in section V.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.161

II. Ant Colony Optimization

Ant colony optimization is a metaheuristic optimization method based on certain behavioural patterns of foraging ants [12]. Ants have shown the ability to find optimal paths between their nests and sources of food. This intelligent path-finding activity is based on stigmergy – indirect communication through modification of the environment. Ants travel randomly to find food, and when returning to their nest, they lay down pheromones. When other foraging ants encounter a pheromone trail, they are likely to follow it. The more ants travel on the same trail, the more intensive the pheromone trace is, and the more attractive it is for other ants.

Emulation of this behaviour can be used as a probabilistic computational technique for solving complex problems that can be cast as finding optimal paths [12]. An artificial ant, k, placed on vertex i, moves to node j, with probability

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in N_i^k} (\tau_{il}^{\alpha}\eta_{il}^{\beta})}, \tag{1}$$

where N_{ik} represents the neighbourhood of node i for ant k (i.e. a set of nodes that are available for the ant to move to), τ_{ij} is the amount of pheromones placed on arc a_{ij}, and η_{ij} corresponds to a-priori information reflecting the cost of passing the arc. After ants finish their forward movement, they return to the nest with food. The tour of ant k is denoted T^k. Its length, C^k, is used to determine the amount of pheromones, $\Delta\tau_{ij}^{k}$, to be placed by the ant on each arc, ij, on the trail that led to the food source

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k} & \text{if arc } (i,j) \text{ belongs to } T^k \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}. \tag{3}$$

Alternatively, $\Delta\tau_{ij}^{k}$ can be derived from the solution quality expressed as the amount of food collected, L_k.

After all ants finish one round of their movement, the amount of pheromones on each arc is reduced through evaporation

$$\tau_{ij} = (1-\rho)\tau_{ij}. \tag{4}$$

The coefficients α, β, and ρ are general parameters of the algorithm that control the ratio between exploitation of known solutions and exploration of new areas of the search space. This canonical form of the ACO algorithm is called Ant System (AS).

There are numerous variants of the ant algorithm. Modifications of the original ant system, such as elitist ant system and ant colony system [12], max-min ant system, fast ant system, ant-Q, and antabu, have been designed and applied in various problem domains, including bioinformatics, scheduling, data clustering, text mining, and robotics [15]. They have also been successfully used for finding optimal paths in complex networks. They perform best when the solved problem has a suitable a-priori heuristic information, and especially when some sort of local search algorithm is employed [12].

A. Formalization and Interpretation of ACO

A number of different formal methods has been used to study and interpret the ant-inspired problem-solving metaphor represented by ACO.

Birattari et al. [3] defined a formal framework named *ant programming* that identified ACO as a generic problem solving strategy based on an incremental Monte Carlo construction of problem solutions biased by a shared memory and linked it to the fields of optimal control and reinforcement learning.

Dorigo and Blum have shown in [9] that ACO can be linked to the stochastic gradient ascent and cross-entropy methods. The authors used an algorithmic framework called *model-based search* to represent the algorithm by its probabilistic model and showed that the problem of finding good parameters of such model is equivalent to solving the underlying combinatorial optimization problem.

An interesting interpretation and analysis of the processes in an ant colony based on the notion of extended cognitive processes (*extended mind*) was proposed by Bosse et al. [4]. The authors connected the principle of stigmergy to the biological concept of external mental states. The ACO processes were in this approach summarized by an ontology of agents' physical and mental states and their relations in time were described by a temporal trace language. Dynamic properties of ACO were studied on several aggregation levels and used e.g. for a simulation-driven system verification.

Guthjar [19] proposed an analytical framework to model and investigate the finite-time dynamics of ant-inspired methods. The framework, defined for algorithms with fitness-proportional pheromone update strategy, was used to construct and prove a limit theorem on the approximation of ACO by a deterministic process based on a system of ordinary differential equations. Moreover, it was shown that the proposed formal model can be used to derive the evolution of expected fitness values.

Another, more general, framework for a unified description of the wide class of stochastic swarm algorithms was presented by Kang et al. [21]. The unified model was applied to ant colony system and several other swarm methods including particle swarm optimization and estimation of distribution algorithms.

Pellegrini and Favaretto [24] proposed a formal description of the exploration conducted by various metaheuristic algorithms in given problem space. The proposed problem and algorithm independent formalism was used to quantify the amount of exploration performed by each investigated method. The study has shown that such formal notion of exploration is connected to the overall performance of the metaheuristic search process. The method was evaluated on an ACO and a genetic algorithm for the TSP and the authors outlined several possibilities how to use the algorithm e.g. for the control and tuning of metaheuristic search and optimization methods.

This short overview demonstrates that there has been a number of different attempts to formalize the intuitive mechanisms of ant-inspired algorithms by various formal methods. In this work, we adopt the complex network model of population dynamics proposed by Zelinka et al. for evolutionary methods [7], [31], [6] and modify it for the use with ant-inspired algorithms for permutation problems.

III. Network Interpretation of Ant Interaction

The main aim of this work is the definition of an intuitive framework for a network interpretation of ant interaction in ant-inspired algorithms. For the sake of simplicity, the method is defined for a distinct class of discrete optimization problems, permutation problems.

The network interpretation of agent interaction has two major parts. First, the amount of interaction between ants in a colony during each iteration of the algorithm is evaluated. Second, the ants' *interaction network* and its dynamic aspects (i.e. updating strategy) are defined.

A. Quantification of Ant Interaction

Consider a permutation problem defined by an undirected graph $G = (V, E)$, with the number of vertices in V denoted by n (i.e. $n = |V|$), and an ant-inspired algorithm with a colony of artificial ants, $\mathcal{A}$. For each ant, $a_i \in \mathcal{A}$, let the permutation of elements defined by its path in G at a time step, t, be called $\pi_i(t) = (\pi_i^t(1), \pi_i^t(2), \dots, \pi_i^t(n))$. For each pair of ants from $\mathcal{A}$, a_i and a_j, $i \neq j$, define the intersection of $\pi_i(t)$ and $\pi_j(t)$ as

$$\pi_i(t) \cap \pi_j(t) = \mathcal{S}(\pi_i(t), \pi_j(t)), \tag{5}$$

where the function $\mathcal{S} : S_n \times S_n \rightarrow \mathbb{R}$ evaluates a *similarity* between the two permutations and S_n is the set of all permutations of length n. The similarity should reflect the amount of information exchanged by the two ants that generated $\pi_i(t)$ and $\pi_j(t)$, respectively. This amount should be proportional to the level of *agreement* between the ants, i.e. to the number of arcs in G where both ants, a_i and a_j, deposited pheromones at the iteration t.

An arbitrary similarity measure can be used to implement $\mathcal{S}$. It should reach its maxima for $\pi_i(t) = \pi_j(t)$ and minima, for $\pi_i(t) \cap \pi_j(t) = \emptyset$, i.e. when both ants chose completely different paths in G. In this work, the bi-directional version of the *adjacency metric* [26], [27] is used as permutation similarity. The adjacency metric counts the number of times elements u and v, $u, v \in \{1, \dots, n\}$, are adjacent to each other in both, $\pi_i(t)$ and $\pi_j(t)$. The bi-directional adjacency relation is defined by

$$adj_\pi(u, v) = \begin{cases} 1, & |pos_\pi(u) - pos_\pi(v)| = 1 \\ 1, & |pos_\pi(u) - pos_\pi(v)| = n \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where $pos_\pi(x)$ is the position of an element, $x \in \{1, \dots, n\}$, in a permutation, π.

Remark 1. *The second case of eq.* (6) *is in effect when u and v are the first and the last element of the permutation, respectively. In such case, $\pi = (u, \dots, v)$ and $adj_\pi(u, v) = 1$. That is a desirable behaviour for the TSP because it seeks for a Hamiltonian* cycle *in a graph.*

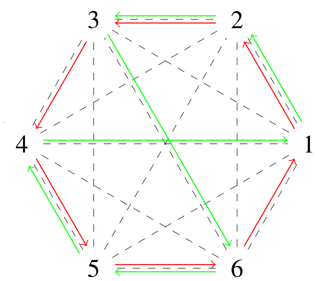


Fig. 1. An example of the similarity measure $\mathcal{S}_{adj}$ for the paths traversed by two ants represented by red and green arrows, respectively. The paths correspond to two permutations, $\pi_i = (1, 2, 3, 4, 5, 6)$ and $\pi_j = (1, 2, 3, 6, 5, 4)$. In this example, the ants have agreed on 4 edges, (1,2), (2,3), (4,5)≡(5,4), and (5,6)≡ $(6, 5)$, so $\mathcal{S}_{adj} = \frac{2}{3}$.

The bi-directional adjacency metric, $\mathcal{S}_{adj}$, is for two permutations of length n, π and σ, defined by

$$\mathcal{S}_{adj}(\pi, \sigma) = \frac{1}{n} \sum_{u,v \in \{1,\dots,n\}} adj_\pi(u, v) \cdot adj_\sigma(u, v). \tag{7}$$

$\mathcal{S}_{adj}$ returns values from the range [0, 1]. Obviously, $\mathcal{S}_{adj}$ is for two identical permutations, $\pi = \sigma$, equal to 1. If π and σ agree on exactly one half of the edges in G, the value of $\mathcal{S}_{adj}$ is 0.5. If π and σ are disjoint, the value of $\mathcal{S}_{adj}$ is 0. A visual example of $\mathcal{S}_{adj}$ for a permutation problem with 6 elements is shown in fig. 1.

Remark 2. *Ant-inspired algorithms are based on the collaboration between agents. In ACO algorithms with a single type of pheromones, all ants are attracted towards an edge with a high amount of pheromones with the same intensity. Intuitively, the value of $\mathcal{S}_{adj}$ will have a growing tendency as the ants will be coming to an agreement on an optimum path in G.*

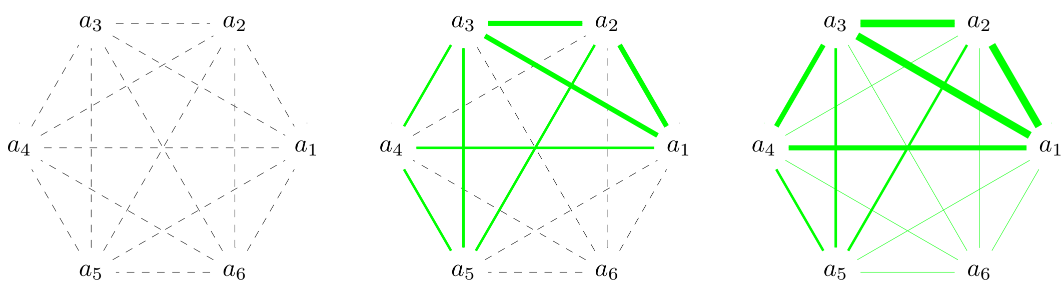
B. Interaction Network

For a permutation problem on G and an ant-inspired algorithm with a colony of m artificial ants, $\mathcal{A}$, consider a dynamic *interaction network*, $\mathcal{I}_G^\mathcal{A} = (V_\mathcal{I}, E_\mathcal{I}, w_\mathcal{I})$. The network is represented by a complete undirected edge-weighted graph on m vertices, $\mathcal{K}_m$, with edge weights evolving in time.

Each vertex, i, in $\mathcal{I}_G^\mathcal{A}$ corresponds to an ant, a_i, in $\mathcal{A}$. The weight of an edge, $e = (ij)$, at time t, $w_{ij}(t)$, reflects the intensity of interaction between ants a_i and a_j in the past t iterations. In every iteration, the weights should be modified so that they increase when the ants reach an agreement on a certain portion of their paths in G and decrease in every other case. This behaviour is analogous to the notion of pheromone deposition and evaporation in ACO or weight increase and forgetting in social network analysis [23].

At time $t = 0$, the initial weights in an interaction network are set to zero:

$$\forall i, j \in E_\mathcal{I} : w_{ij}(0) = 0. \tag{8}$$



(a) t = 0. (b) t = 10. (c) t = 100.

Fig. 2. An example of interaction network for a colony with 6 artificial ants. Different subfigures show the network at different iterations. Edge thickness is in all graphs proportional to edge weight. Note that this is an illustration of the desired behaviour and the structure and properties of real-world interaction networks depend, among others, on the choice of $\mathcal{S}$ and Δ, respectively.

The general weight updating scheme considers increase and decay of edge weights.

$$w_{ij}(t+1) = \Delta(w_{ij}(t)) + \mathcal{S}(\pi_i(t+1), \pi_j(t+1)), \quad (9)$$

were $\mathcal{S}$ is a permutation similarity measure and $\Delta : \mathbb{R} \rightarrow \mathbb{R}$ is an arbitrary decay function. In this work, we use the similarity measure $\mathcal{S}_{adj}$ defined in eq. (7) and a simple constant decay function

$$\Delta_\rho(x) = \max(0, x - \rho), \quad (10)$$

where $\rho \in \mathbb{R}$ is a parameter determining the rate at which past agent interactions are forgotten in every iteration. An illustration of an interaction network for a colony of 6 ants at different stages of the algorithm is shown in fig. 2.

Remark 3. *The constant decay function, Δ_0, can be with $\rho = 0$ used to model an interaction network with* non-decreasing *edge weights. Such network never forgets an interaction between any pair of ants.*

C. Summary

A conceptual framework for the interpretation of ant interaction in ACO algorithms from a complex network perspective was outlined in this section. The general concept includes a mechanism for the quantification of ant interaction, defined by eq. (5), and a set of generic rules for the construction and updates of an interaction graph described by eq. (8) and eq. (9), respectively.

Equation (6) and eq. (7), on the other hand, provide a particular implementation of the similarity function and a simple weight decay rule is proposed in eq. (10). These straightforward functions are suitable for the use with simple permutation problems such as the TSP instances from the TSPLIB95 library[1]. These functions are utilized in section IV to illustrate the proposed complex network interpretation of ant interaction.

[1]http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

IV. Network Model of Ant Interaction in the TSP

This section provides a concrete illustration of the network interpretation of ant interaction. Several instance of the traveling salesman problem from the well-known TSPLIB95 library are solved by an ACO algorithm and the corresponding interaction networks are constructed, visualized, and investigated.

A. Traveling Salesman Problem

The traveling salesman problem is an iconic hard combinatorial optimization problem with a long tradition, a number of different variants, and countless applications [1], [25]. Informally, the TSP consists in finding the shortest (least expensive) route between n cities. In mathematical terms, the TSP looks in a weighted graph, $G = (V, E, w)$, with a set of vertices V, number of vertices $n = |V|$, set of edges E, and a set of edge weights $w = \{w_e \in \mathbb{R} \mid \forall e \in E\}$, for a Hamiltonian cycle, $\mathcal{L}$, with a minimum sum of weights on the edges of the cycle [25].

A particular instance of the TSP is often represented by a cost matrix, $C^{n \times n} = (c_{ij})$, where $c_{ij} = w_e$ for all edges $e = (ij)$. If G is not complete, missing edges can be in C represented by an arbitrary *large weight*.

The TSP can be also formulated as a permutation problem [25]. For a set of all permutations of a set of n objects, S_n, find $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ such that the cost of the permutation (i.e. objective function),

$$f_{\text{obj}}(\pi) = c_{\pi(n)\pi(1)} + \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}, \quad (11)$$

is minimized.

The TSP is of paramount significance for the ACO community. It was among the first problems used to demonstrate the concepts and usefulness of ant colony optimization during the advent of ant-inspired computing [10], [11]. Later on, it was frequently used as a benchmark problem to demonstrate the effectiveness of new algorithm improvements [13], [11], innovative ACO variants [5], [17], [29], and novel software and hardware solutions [2], [14]. The TSP is in this work used to illustrate the proposed concepts of network interpretation

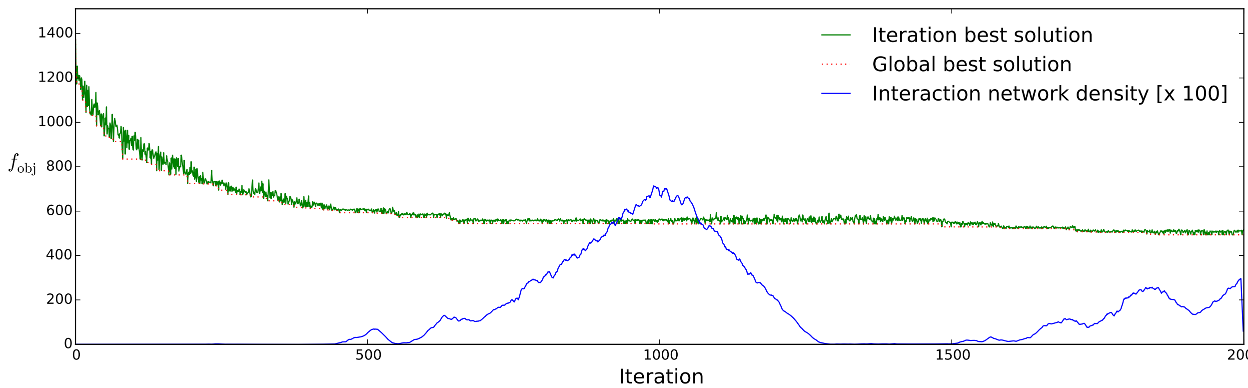
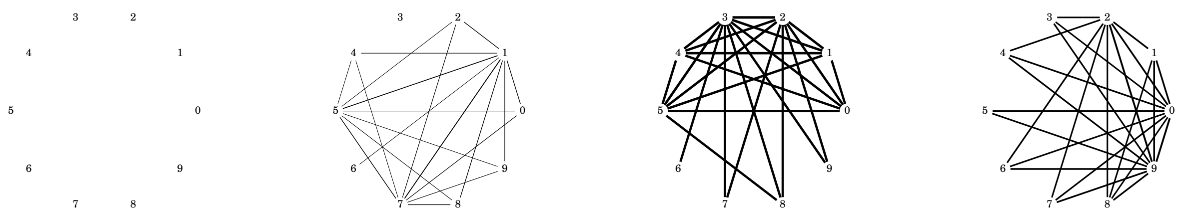


Fig. 3. An example ACO run for the *eil51* TSP instance.



(a) Iteration 0. (b) Iteration 500. (c) Iteration 1000. (d) Iteration 2000.

Fig. 4. Complex network model of an ACO solving *eil51* at different iterations.

of ant interaction because of its familiarity, the simplicity of TSPLIB95 instances, and known best solutions.

B. Interaction Network for the TSP

To demonstrate the proposed network model of ant interaction, an elitist ant system [12] was implemented and employed to solve TSP instances from the TSLPIB95 repository. The algorithm utilized a-priori information $\eta_{ij} = c_{ij}$, and pheromone amplification rate, α, and a-priori information amplification rate, β, set to 1. The amount of deposited pheromone was proportional to the quality of solutions and the evaporation rate, ρ, was set to 0.1. The number of artificial ants, m, was fixed to 10 and the algorithm was executed for 2000 iterations. The selected ACO variant and its parameters are based on best practices, authors' past experience, and extensive experimental trial-and-error runs. The number of ants was selected so that the algorithm was able to find a reasonable problem solution and the network model remained clear and comprehensible. We note that the goal of presented computational experiments was not the search for optimum TSP solutions but the illustration of the network interpretation of ant interaction on a real-world problem.

The network model of ant interaction utilized the bi-directional adjacency metric, $\mathcal{S}_{adj}$ (7), and constant decay function, Δ_ρ (10), with the parameter ρ equal to 0.9. It means that 90% of a full interaction between each pair of ants was forgotten by the network in every iteration. Graph *density*, $den(G)$, was computed for $\mathcal{I}_G^{\mathcal{A}}$ in every iteration to provide an initial assessment of its properties [16]

$$den(\mathcal{I}_G^{\mathcal{A}}) = \frac{2 \cdot \sum_{(ij) \in E_{\mathcal{I}}} w_{ij}}{m(m-1)}. \tag{12}$$

Visual descriptions of two ACO runs solving two particular TSP instances, *eil51* and *st70*, are provided in fig. 3 and fig. 5, respectively. Although only two particular algorithm runs for two TSP instances are shown, the displayed behaviour is typical and was observed also in other independent ACO runs and for other TSP instances. The figures show for each iteration current (i.e iteration-best) and so far best (i.e. global-best) problem solutions as well as the running mean of the density of the interaction network. Running mean with window size 5 was computed to smooth out sudden changes of this measure and to emphasize its trends.

It can be immediately seen that the values of $den(\mathcal{I}_G^{\mathcal{A}})$ correspond to the trends of the objective function, f_{obj}. The density is zero or decreasing when the values of f_{obj} are improving and increasing when the algorithm becomes stuck in a local minima and f_{obj} stagnates. Such situations are characterized by long plateaus on the plot of objective function values. Figure 3 and fig. 5 suggest that the value of $den(\mathcal{I}_G^{\mathcal{A}})$

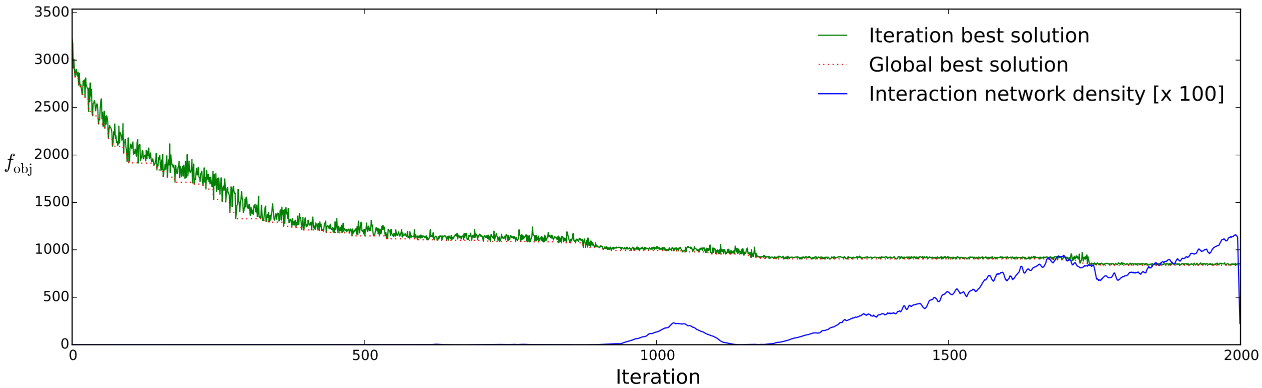
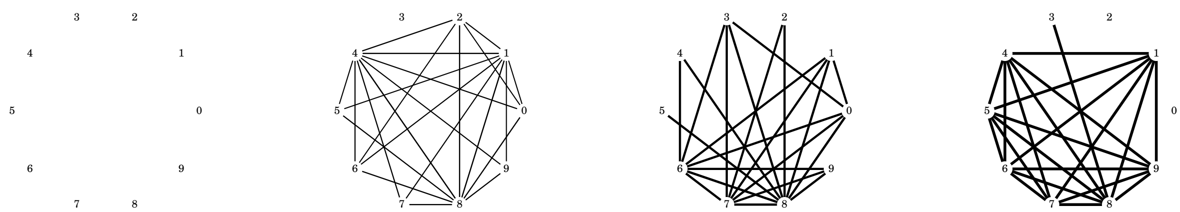


Fig. 5. An example ACO run for the *st70* TSP instance.



(a) Iteration 10. (b) Iteration 100. (c) Iteration 500. (d) Iteration 2000.

Fig. 6. Complex network model of an ACO solving *st70* at different iterations.

drops when the algorithm becomes more explorative, i.e. when the values of the iteration-best solution become increasingly oscillating.

The interaction networks, sampled at different ACO iterations, are for each presented ACO run shown in fig. 4 and fig. 6, respectively. The thickness of edge weights represents the intensity of interaction between corresponding ants. All interaction graphs were drawn using a straightforward circular layout and the location of the vertices has no relation to the solved problem or interaction network state.

V. Conclusions

This work proposes a framework for a complex network interpretation and modelling of agent interaction in ant-inspired algorithms for permutation problems. It allows representing the dynamics of an ant colony optimization algorithm by an dynamic graph that can be subsequently studied and eventually used to control the algorithm. The framework is presented in general terms and its simple implementation for a well-known permutation problem, the travelling salesman problem, is outlined.

Selected TSP instances are in this study solved by an elitist ant system and the corresponding interaction networks are created and analyzed. The conducted computational experiments illustrate the proposed concepts and confirm that the graph-theoretical properties of the interaction network can provide a valuable information regarding the status of the algorithm. The initial experiments show that interaction graph density can be used to identify the state of the algorithm. A steadily increasing value of interaction graph density can be interpreted as a sign of local minima whereas its drop can be seen as a signal that the algorithm has escaped from such a location.

Network interpretation and analysis of ant-inspired algorithms is a promising new field of study that opens a number of research questions. Besides a generalization of the proposed approach to other than just permutation problems (e.g. subset selection [20] or routing [22] problems) and to other types of multiagent and swarm algorithms, the study of the relations between various graph-theoretical properties of the interaction networks and their implications for the optimization procedures is of utmost interest.

Last but not least, network-based modelling and analysis of ant-inspired algorithms is interesting also from a broader perspective since it is now well-understood that real-world ant colonies are governed by networks of interaction [18] and complex network models may contribute to accurate emulation of their behaviour.

Acknowledgement

This work was supported by the Czech Science Foundation under the grant no. GACR P103/15/06700S and partially supported by grants of SGS, VŠB - Technical University of Ostrava, no. SP2015/146 and SP2015/142.

References

[1] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2007.

[2] M. Behalek, S. Böhm, P. Krömer, M. Surkovský, and O. Meca. Parallelization of ant colony optimization algorithm using kaira. In *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*, pages 510–515, 2011.

[3] M. Birattari, G. Di Caro, and M. Dorigo. Toward the formal foundation of ant programming. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Ant Algorithms*, volume 2463 of *Lecture Notes in Computer Science*, pages 188–201. Springer Berlin Heidelberg, 2002.

[4] T. Bosse, C. M. Jonker, M. C. Schut, and J. Treur. Simulation and analysis of a shared extended mind. *SIMULATION*, 81(10):719–732, 2005.

[5] S.-C. Chu, J. F. Roddick, and J.-S. Pan. Ant colony system with communication strategies. *Information Sciences*, 167(14):63 – 76, 2004.

[6] D. Davendra, I. Zelinka, M. Metlicka, R. Senkerik, and M. Pluhacek. Complex network analysis of differential evolution algorithm applied to flowshop with no-wait problem. In *Differential Evolution (SDE), 2014 IEEE Symposium on*, pages 1–8, Dec 2014.

[7] D. Davendra, I. Zelinka, R. Senkerik, and M. Pluhacek. Complex network analysis of evolutionary algorithms applied to combinatorial optimisation problem. In P. Krömer, A. Abraham, and V. Snášel, editors, *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, volume 303 of *Advances in Intelligent Systems and Computing*, pages 141–150. Springer International Publishing, 2014.

[8] G. Di Caro, F. Ducatelle, and L. M. Gambardella. Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks. In N. Sala and F. Orsucci, editors, *Reflecting interfaces: the complex coevolution of information technology ecosystems*, pages 185–216. Idea Group, Hershey, PA, USA, 2008.

[9] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(23):243 – 278, 2005.

[10] M. Dorigo and L. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1):53–66, Apr 1997.

[11] M. Dorigo and L. M. Gambardella. Ant colonies for the travelling salesman problem. *Biosystems*, 43(2):73 – 81, 1997.

[12] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[13] I. Ellabib, P. Calamai, and O. Basir. Exchange strategies for multiple ant colony system. *Information Sciences*, 177(5):1248 – 1264, 2007. Including: The 3rd International Workshop on Computational Intelligence in Economics and Finance (CIEF2003).

[14] W. Elloumi, H. E. Abed, A. Abraham, and A. M. Alimi. A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP. *Applied Soft Computing*, 25(0):234 – 241, 2014.

[15] A. Engelbrecht. *Computational Intelligence: An Introduction, 2nd Edition*. Wiley, New York, NY, USA, 2007.

[16] K. Erciyes. *Complex Networks: An Algorithmic Perspective*. Taylor & Francis, 2014.

[17] C. García-Martínez, O. Cordón, and F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180(1):116 – 148, 2007.

[18] D. M. Gordon. *Ant Encounters: Interaction Networks and Colony Behavior*. Primers in complex systems. Princeton University Press, 2010.

[19] W. Gutjahr. On the finite-time dynamics of ant colony optimization. *Methodology and Computing in Applied Probability*, 8(1):105–133, 2006.

[20] M. Kabir, M. Shahjahan, and K. Murase. An efficient feature selection using ant colony optimization algorithm. In C. Leung, M. Lee, and J. Chan, editors, *Neural Information Processing*, volume 5864 of *Lecture Notes in Computer Science*, pages 242–252. Springer Berlin Heidelberg, 2009.

[21] Q. Kang, J. An, L. Wang, and Q. Wu. Unification and diversity of computation models for generalized swarm intelligence. *International Journal on Artificial Intelligence Tools*, 21(03):1240012, 2012.

[22] P. Krömer, J. Martinovic, M. Radecký, R. Tomis, and V. Snásel. Ant colony inspired algorithm for adaptive traffic routing. In *Third World Congress on Nature & Biologically Inspired Computing, NaBIC 2011, Salamanca, Spain, October 19-21, 2011*, pages 329–334, 2011.

[23] M. Kudělka, Z. Horák, V. Snášel, P. Krömer, J. Platoš, and A. Abraham. Social and swarm aspects of co-authorship network. *Logic Journal of IGPL*, 20(3):634–643, 2012.

[24] P. Pellegrini and D. Favaretto. Quantifying the exploration performed by metaheuristics. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(2):247–266, 2012.

[25] A. Punnen. The traveling salesman problem: Applications, formulations and variations. In G. Gutin and A. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*, pages 1–28. Springer US, 2007.

[26] C. Reeves. Landscapes, operators and heuristic search. *Annals of Operations Research*, 86(0):473–490, 1999.

[27] T. Schiavinotto and T. Stützle. A review of metrics on permutations for search landscape analysis. *Comput. Oper. Res.*, 34(10):3143–3153, Oct. 2007.

[28] T. Stützle and H. Hoos. Improvements on the ant-system: Introducing the max-min ant system. In *Artificial Neural Nets and Genetic Algorithms*, pages 245–249. Springer Vienna, 1998.

[29] C.-F. Tsai, C.-W. Tsai, and C.-C. Tseng. A new and efficient ant-based heuristic method for solving the traveling salesman problem. *Expert Systems*, 20(4):179–186, 2003.

[30] A. C. Zecchin, A. R. Simpson, H. R. Maier, A. Marchi, and J. B. Nixon. Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem. *Water Resources Research*, 48(9):n/a–n/a, 2012. W09505.

[31] I. Zelinka, D. Davendra, J. Lampinen, R. Senkerik, and M. Pluhacek. Evolutionary algorithms dynamics and its hidden complex network structures. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 3246–3251, July 2014.