ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 2, No. 1, 2017

CYBER-PHYSICAL SYSTEM FOR SOLVING TRAVELLING SALESMAN PROBLEM

Oleksandr Muliarevych

Lviv Polytechnic National University, 12, S. Bandera str., Lviv, 79013, Ukraine
Author e-mail: *mvalex@i.ua*

Submitted on 19.09.2017

Abstract: **In this paper new approaches for solving dynamic Travelling Salesman Problem (TSP) in conditions of partly unknown input data are given.**

Index Terms: **agents, TSP, ant colony algorithm, swarm behavior, distributed TSP, network routing, cyber-physical systems, robotics.**

INTRODUCTION

In the report new approaches for solving one of the combinatorial optimization tasks – the Travelling Salesman Problem (TSP) [1] using developed system modeling based on collective agents' swarm [2] behavior as a base are established. For creation of swarm behavior model with indirection communication between agents was used ant colony algorithm with its developed modifications. Developed approaches enable to solve dynamic asymmetrical TSP (DATSP) in conditions of partly unknown input data, which is one of the important tasks [3]. It is related to practical usage: routing tasks in a real-life computer networks. Search for approach of solving distributed TSP in the same conditions is another corner case, which adds to DATSP definition condition of distribution of point for visit by agents. Solving of distributed TSP is an important task in logistic cyber-physical systems, in cyber-physical robotics [4] of space, military or research proposals.

PROBLEMS OVERVIEW

Search for route tasks in a real-life communication network is related to Dynamic Asymmetrical TSP (DATSP) in conditions of partly unknown input data. In communication network of typical cyber-physical system when agents are located on any node (P(i)), it is possible to retrieve only accessible from current node information about available nodes in network and transfer cost to them: P(i) → P(k), where k – index in array of nodes with size M, which could be accessed from current node (i). Such conditions disable guarantees of TSP solving, the whole network state monitor is unavailable – available is the data accessible from initial location – start node. Along the iteration of routes search by ants-agents cycle, from start of agents' travel till the end of it in initial node there is no information about TSP solving process. Therefore, input data is not only dynamic but also partly unknown along the whole calculation process. Besides showing possibility and performance of developed cyber-physical system for static TSP solving, main researches were focused on solving dynamic TSP and DATSP in conditions of partly unknown input data.

To use the developed system [3, 5] for solving routing tasks in real-life computer network into SSTSP [1] formulation the following conditions should be applied: 1) input data dynamic changes; 2) no existing TSP solution possibility (route to visit all nodes, visit each node one time); 3) a partly unknown input data – each agent has an access to information about movement possibilities to another nodes and single costs accessible from initial node; 4) asymmetry – movement costs between any of the network's nodes in different directions is not equal. All described conditions are unable to solve the task by existing developed system.

During the investigations the following problems were found: 1) Possibility to get into "Deadlock" and "Trap" situations. 2) Impossible update of classical digital mark values ("pheromone update" in ACO), which is executed after the return of all agents to the initial node. 3) Real-time routes usage. Typically, in existing cyber-physical systems for TSP solving, results are used after being calculated. Possible dynamic changes and a partly unknown input data make usual approach inefficient. 4) Infinite increase of digital mark values, which are placed on network links between nodes – endless "ant colony memory" strengthening. Let's have an overview of each of the described problems and possible solutions on high-level details. Problems that appear during DATSP solving in conditions of partly unknown input data, need to be solved.

Developed decentralized approach of DATSP solving can be extended into distributed TSP solving based on agent indirect communication via node as arbitrary, which makes decision which agent will pass to it. Accept decision module is based on theory of games [6]. Autonomous agents, who take part in TSP solving, could get into "conflict" situation, when several of them want to move to the same node at the same time. If node was visited by any agent it should not be visited anymore.

Self-organized distribution of agents between nodes, during decision point in a "conflict" situation can be done using one of the strategies: 1) ‘cooperation’ between agents; 2) ‘conquest’ between agents. Both of those strategies could be combined with constraints from real-life requirements of practical task solving.

DEVELOPED APPROACHES

Using methods of developed classification [3] for TSP solving the analysis of existing methods was made. According to the results of analysis the most perspective for solving dynamic TSP methods with agents’ collective usage was selected. Developed computer system is based on ACO algorithm usage with implemented and proposed algorithms modifications: initialization phase and decision procedure improvement [5].

It consists of two subsystems: graphical for resulting routes representation and result TSP route calculation. During TSP route calculation “Search routes by ants-agents cycle” is executed, which is based on ACO algorithm with proposed modifications.

Decision module is used during simulation of ants-agents’ behavior and so called “ant colony memory” – system of marks – digital values (pheromone in nature ants), which represent common agents’ memory, stored and modified by them. Based on this memory and real-life situation ants-agents are orienting during “definition of next to go node” procedure. Each digital value of mark is related to link between nodes and set in accordance with agents’ experience. Digital mark value is updated after it was visited by ant-agent. Represented on schema additional program module is based on local optimization methods that are used for retrieved resulting routes improvement.

Implementation of additional program module, which is based on local optimization methods usage, into developed computer system for solving TSP enables to reduce difference between quasi-optimal and optimal routes, as well as the number of iterations in cycle of routes search by ants-agents. Additional program module can process resulting routes in 3 possible modes, using 2-opt, 2.5-opt or 3-opt local optimization methods [7]. Also, this module can be switched off.

Mode of optimization is configured from beginning, but during implementation was changed to dynamic value, that is automatically defined by cyber-physical system using evaluated frequency of dynamic changes of system, which is done in environment evaluation module, that is included in software installed on every node of communication network.

In terms of research and implementation results evaluation of additional program module, that is based on local optimization methods and used in developed computer system, library file att532.tsp [8] was used. Library includes all input data for TSP of 532 points (USA cities coordinates). During TSP solving process by ant colony optimization method the digital mark values are used – “ant colony memory”, which can be analyzed. Digital marks existence enables easy adaptation of agents to dynamic input data and environment conditions change during the process of TSP solving. “Ant colony memory” overview for TSP of 532 points is shown on Fig. 2. Link width between nodes is set in accordance with digital mark value (“pheromone” level at biological ants as analogue) placed on it.

On Fig. 2 “ant colony memory” state after 10 iterations execution of routes search by ants-agents cycle is shown. “Ant colony memory” state without local optimization methods usage is demonstrated on Fig. 2, a. On Fig. 2, b “ant colony memory” is demonstrated after TSP solving with 3-opt local optimization algorithm usage. As we can see, usage of local optimization methods boosts disappearance of additional edges process. Extra links between nodes from not optimal routes become unsuitable and will be excluded from usage, because of digital mark values reducing.

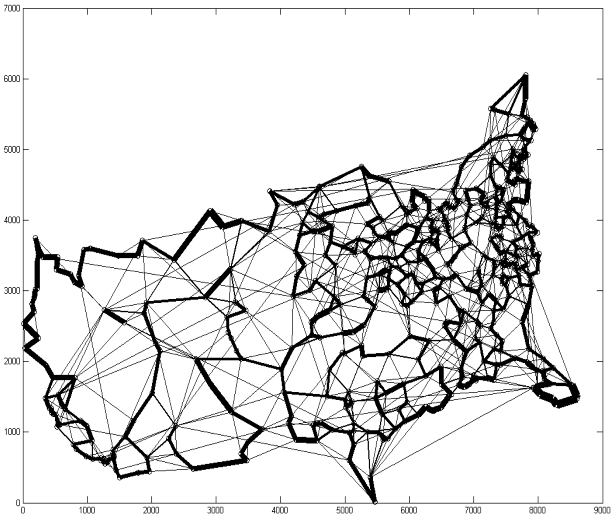


Fig. 2a. Collective memory (“ant colony memory”) overview – digital mark values matrix in process of solving TSP (532 points) without local optimization methods usage

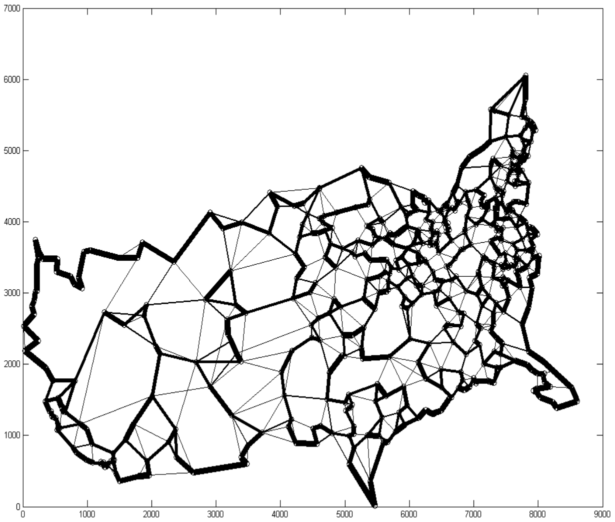


Fig. 2b. Collective memory (“ant colony memory”) overview – digital mark values matrix in process of solving TSP (532 points) with 3-opt local optimization algorithm usage

In Table 1 calculated resulting routes during first 3 iterations of search routes by ants-agents cycle in different modes could be found: 1) without local optimization methods usage; 2) with 2-opt, 2.5-opt, 3-opt implemented local optimization methods usage after each iteration of cycle. Average calculation time spent for TSP solving in observed and developed computer systems functional modes is shown in Table 1.

Table 1

Calculation process of TSP (532 points)

	Length of route				Improvements \ permutations count			Average calculation time, (s)			
I	without local optimiza-tion methods usage	with local optimization methods usage			2-opt	2,5-opt	3-opt	without local optimi-zation methods usage	with local optimization methods usage		
		2-opt	2.5-opt	3-opt					2-opt	2.5-opt	3-opt
1	49787	30105	29387	28780	428	1239	614	0.9	4.4	4.6	4.95
2	39067	29806	29327	28675	202	1012	625				
3	38939	29665	29202	28375	224	797	625				

Optimal route length for current TSP is 27686 (based on data from library file). Optimal route was found by developed computer system in 3-opt local usage mode optimization method. Each local optimization method usage, depending on its complexity, increase result routes calculation time, but highly decreases number of iterations of routes search by ants-agents cycle and increase accuracy of retrieved result. In Fig. 3 TSP of 532 points solutions are graphically shown. Resulting routes were achieved in four functional developed system modes, after 30 iterations of routes search by ants-agents cycle: without local optimization methods usage (Fig. 3,a); with 2-opt local optimization method usage (Fig. 3, b); with 2.5-opt local optimization method usage (Fig. 3, c); with 3-opt local optimization method usage (Fig. 3, d).

Length of resulting route in accordance with table and figures above is reduced proportionally to local optimization method complexity: higher complexity level of used method – shorter length of resulting route. Developed computer system based on swarm intelligence agents' behavior model, even applying local optimization methods, can solve tested TSP (532 points) in lower time than LKH [7] application, which requires approximately 7 seconds for tested TSP solving.

Developed computer system can output quasi-optimal resulting route already in approximately 1 second of calculation and difference from optimal result is less than 6 %. Constructive type of used ant colony optimization algorithm retrieves possible route even after the first iteration of routes search by ants-agents cycle.

Accordingly retrieved quasi-optimal resulting route for TSP of less than 6000 nodes could be achieved already after several milliseconds after calculation process was started. This result could be sent to additional program module, based on local optimization methods, for further length decrease of retrieved route.

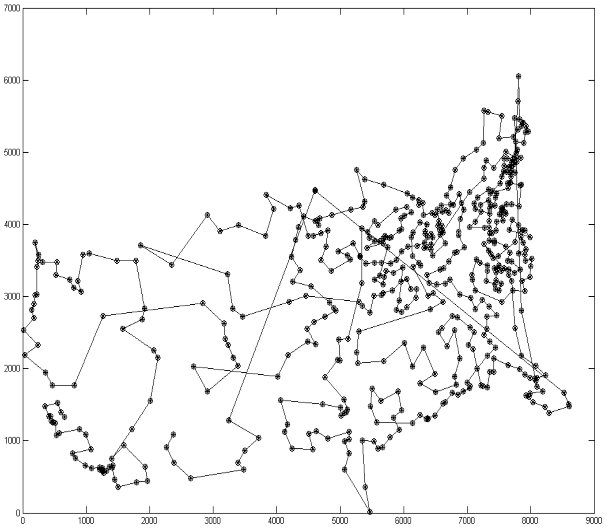


Fig. 3a. Resulting routes for tested TSP (532 points), without local optimization methods usage

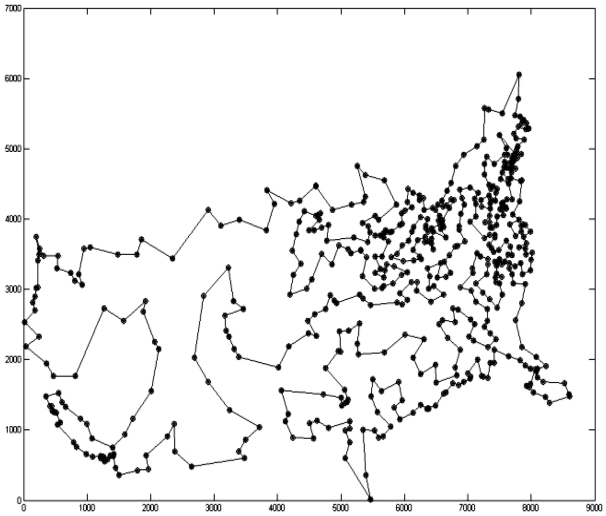


Fig. 3b. Resulting routes for tested TSP (532 points), with 2-opt local optimization method usage

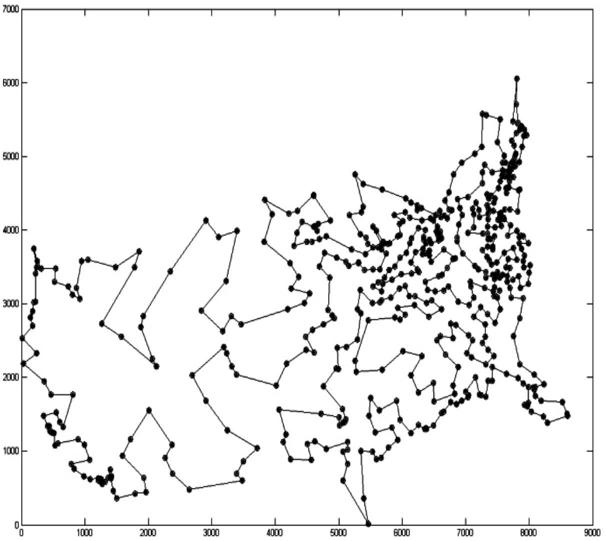


Fig. 3c. Resulting routes for tested TSP (532 points), 2.5-opt local optimization method usage

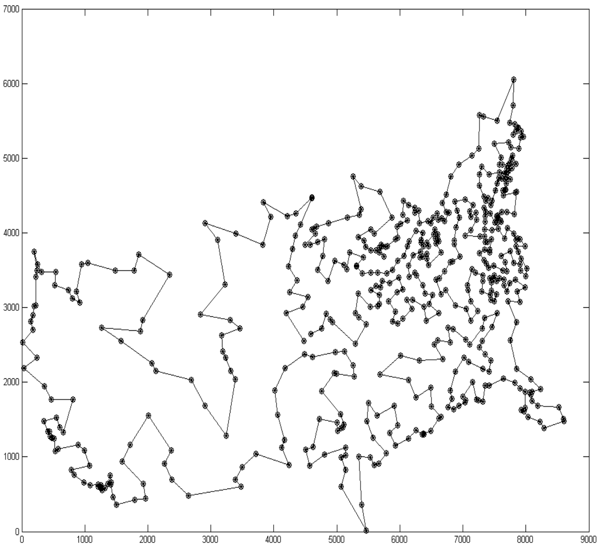


Fig. 3d. Resulting routes for tested TSP (532 points), 3-opt local optimization method usage

Along the whole research within developed computer system with additional program module several TSP instances from the library [Heidelberg University from 1995] were used. Results are represented in Table 2.

Highlighted results are the optimal for related TSP. If optimal length of the route is not achieved, it is added in brackets. The results above were retrieved within 3-opt local optimization method mode. In accordance with results, developed computer system with additional program module can find quasi-optimal route with less than 3 % difference from optimal result for TSP with size less than 6000 nodes.

CRITICAL SITUATIONS PROCESSING

As a result of partly unknown input data and its dynamic changes input TSP task is unable to be solved. In other words, visit of all network nodes could be disabled because of a specific network structure: inaccessible nodes or links between them.

Mentioned conditions for agents create a chance to get into one of the following critical situations:

a) “TRAP” situation is when ant-agent gets into network node after data processing procedure and discover that node is disconnected from the network (there are no other accessible nodes to continue route). Agent gets into “trap” on current node.

b) “DEADLOCK” situation is when ant-agent gets into endless cycle while travelling in network.

“Trap” situation is shown in Fig. 4. Having moved from node B to node A, ant-agent explores that movement back to node B or to next node is impossible. It means that ant-agent gets into “TRAP” situation on node A.

Proposed steps to overcome this situation in scope of developed system:

A.1) Set status of mark as “TRAP” on current node to notify and save other ants-agents. This status will be removed in case of agent is able to move to next node or is destroyed (”die”).

A.2) Try to find accessible nodes. If it is possible to continue route and remove “TRAP” status after successful movement to another node. If there are no accessible nodes, then go to step A.3.

A.3) Wait for a set amount of time and go to step A.2. It is possible to set limit of such attempts to exit. After all attempts were checked, “death” of ant-agent is caused (its self-destructions) and “TRAP” status will be removed on current node.

Table 2

Solution for TSP instances from the library

TSP name	Best result	Worst result	Average result	Average iteration count	Average total time, s	Quasi-optimal result achieve-ment time, s
a280	**2579**	**2579**	2759.00	28.50	6.59	0.29
ali535	**202339**	202454	202390.10	482.90	24.02	12.02
att532	**27686**	27705	27698.40	396.10	18.63	7.61
att48	**10628**	**10628**	10628.00	3.50	1.95	0.01
berlin52	**7542**	**7542**	7542.00	1.00	2.17	0.006
bier127	**118282**	**118282**	118282.00	17.10	5.47	0.13
ch130	**6110**	**6110**	6110.00	10.90	4.81	0.08
ch150	**6528**	**6528**	6528.00	15.20	4.93	0.11
d198	**15780**	15781	15780.40	348.60	8.66	3.06
d493	35004 (35002)	35033	35015.80	548.60	19.78	11.05
d657	48930 (48912)	48989	48967.00	563.20	27.07	15.50
d1291	**50801**	50825	50814.80	350.30	40.17	14.87
d1655	62169 (62128)	62411	62286.40	857.50	51.94	45.14
dsj1000	18665854 (18660188)	18699764	18689299.20	866.20	48.92	42.78
eil51	**426**	**426**	426.00	2.90	1.75	0.01
eil76	**538**	**538**	538.00	7.20	2.51	0.03
eil101	**629**	**629**	629.00	12.40	3.29	0.06
fl417	11861	11861	11861.00	302.90	14.88	4.65
fl1400	20188 (20127)	20259	20220.80	715.10	60.62	43.25
fl1577	22329 (22249)	22485	22360.80	599.70	46.75	28.10
gil262	**2378**	**2378**	2378.00	72.90	8.84	0.78
gr96	**55209**	**55209**	55209.00	8.80	3.90	0.04
gr137	**69853**	**69853**	69853.00	12.10	5.30	0.09
gr202	**40160**	**40160**	40160.00	124.00	8.82	1.17
gr431	**171414**	171463	171426.80	588.80	18.11	10.90
gr666	**294382**	**295059**	294584.50	565.30	27.21	15.60
lin318	**42029**	**42029**	42029.00	405.30	11.86	4.91
nrw1379	56717 (56638)	56812	56762.60	848.90	53.68	46.91
pr2392	378599 (378032)	379939	379173.70	752.00	91.42	71.59
rl5915	568473 (565530)	576503	571753.00	900.20	366.64	330.33

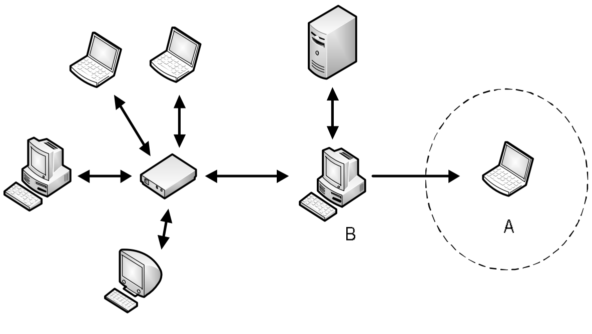


Fig. 4. “TRAP” situation on node A

Computer network with possible cases for “Deadlock” situations are shown in Fig. 5.

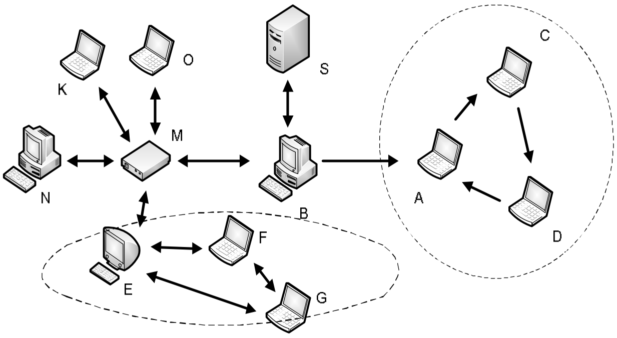


Fig. 5. “DEADLOCK” situations

Proposed steps to overcome this situation in scope of developed system:

B.1) Store IDs list of visited nodes – “taboo list” with time sequence of these visits during solving TSP – walked route;

B.2) Choose the next node to visit from pool of not visited by current time nodes. If described nodes were not found, go to step B.3;

B.3) Search for accessible start node P(0) aiming to finish the route. If it is not found, then go to step B4;

B.4) Find next node to move P(i+1) from visited nodes pool, which matches to the following condition:

$$P(i+1) \neq P(i-1), \qquad (1)$$

where P(i-1) is previous node in route.

Therefrom, short “deadlock” consisting of 2 nodes could be avoided. If node wasn't found, wait until a set amount of time is over and go to step B.2. If node was found, go to step B.5;

B.5) Check whether selected node in step B.4 is not a “deadlock” cycle part: analysis of common link sequence in traveled route aiming to find repeated parts. If agent’s presence in “deadlock” was found then continue next movement forward aiming to find node Pk, which will not be related to nodes pool included in “deadlock”. Thereby, if presence in “deadlock” during check was not found, then move to this node, otherwise return to step B.4 if remained nodes have not been checked yet. If all nodes are checked, then move to found node and go to step B2.

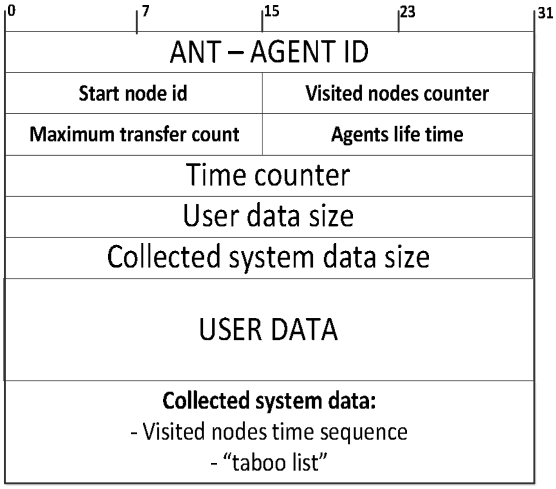
In ACO classical marks update – “pheromone update” (pheromone covering and evaporation) procedure was executed after the end of route and return to the start position of all agents [2]. It enables to increase mark values in accordance with calculated result routes costs. In case of frequently dynamic changes of input data, resulting routes could be inappropriate or even impossible to use on practice, because state of network could be changed and new route search will be needed.

For solving this problem new approach was developed: now digital mark values are increased during procedure of agent movements between nodes in moment when ant-agent reaches destination node. The idea of such approach is based partly on ants’ nature, as they lay pheromone (increase value of mark) during all their way. Additionally, for giving benefits to the best-found solutions, after ants-agents have returned to the start node, new defined agents are sent. The aim of these special agents is an attempt to repeat found quasi-optimal routes of lowest costs with an aim to increase additionally digital mark values on visited links. Such special agents return is not compulsory, but successful end of their travel enables reinforcement of the best currently found routes.

Most existing cyber-physical systems for solving TSP [4, 7] use the following approach: 1) Solve of TSP procedure execution in accordance with input data: 2) Result (quasi-optimal route) is used for analysis or possible practical task. In terms of computer network practical task will be data transportation (wideband data transfer). In case of input data dynamic changes, during ants-agents traveling process inside of the network the following situation could appear: the walked links could become inaccessible or transfer cost by them is changed. Such situation makes found routes inefficient for data transfer.

Approach of developed system for TSP solving was changed to solve this problem. Proposed new approach consists of resulting routes calculation process and practical task execution (data transfer) union. It means, that during data transfer, routes search by ants-agents logic is executed aiming to find best (minimal by cost) routes and minimize expenses.

In scope of such approach, found routes are actual in a moment of their calculation and after ants-agents return to the start node – routes search by ants-agents cycle finished. Collected information about network is available for further analysis. Input data for sending is divided into data blocks per ants-agents count and data size. If some agents do not return until defined period because of described “trap” or “deadlock” situations, lost data blocks could be resent by new agents. For this approach agent was redefined. Ant-agent now is a data package; its structure is shown in Fig. 6.



ANT – AGENT ID	
Start node id	Visited nodes counter
Maximum transfer count	Agents life time
Time counter	
User data size	
Collected system data size	
USER DATA	
Collected system data: - Visited nodes time sequence - “taboo list”	

Fig. 6. Ant-agent – data package

Sizes of data package fields in bits are shown in digits on top of Fig.6. Besides user data, package includes the following fields: ant-agent id; start node id (id of node that initiate data send procedure); visited nodes counter; time counter; maximum transfer count (moves between nodes count) and agents' life time (time after which agent will be self-destroyed), that limit time of ants-agents' existence; collected (during calculation) system data: visited nodes time sequence and "taboo list" – list of forbidden to visit nodes.

During developed system for solving TSP investigation and analysis another challenge – endless increase of digital mark values placed on network nodes. Each mark value is related to link between nodes. Digital mark value represents benefits of link between nodes in accordance with collected agent collective's experience. Bigger mark value related to link between nodes enables to choose this link by ants-agents for travel. After agents found quasi-optimal route in conditions, when dynamic changes have no place or have no influence on found results for a long time, digital mark values are repeatedly increased (”pheromone update” procedure in ACO) on the same links, related to found route. Thus, permanently growing up digital mark values on these links in conditions of their accessibility are not able to continue search for other alternative links and routes, possibly more optimal. It means that stored "ant colony memory", that includes found quasi-optimal route, can lose its flexibility and get into endless strengthening process.

To solve this problem new approach was proposed: add new adaptive top limit for mark values Smax – maximum digital mark value. Digital mark value Sij on the link between nodes i and j is defined as:

$$\mathrm{Sij} = T * \mathrm{Vij}, \tag{2}$$

where T – data transfer through network time during TSP calculation process; Vij – digital mark value accumulation speed between i and j nodes.

Assume that data transfer time in conditions, when dynamic input data changes have no place, achieves a huge value T → ∞; then in accordance with expression (2) digital mark value increase infinitely Sij → ∞.

Based on proposed approach with adaptive top limit for digital mark value Smax usage expression (2) could be represented in the following way:

$$\mathrm{SIJ} = \mathrm{MIN}(T * \mathrm{VIJ}, \mathrm{SMAX}). \tag{3}$$

Adaptive top limit for digital mark value Smax is set depending on the following factors: count of agents – N; digital mark value accumulation speed Vij – difference between increasing digital mark value (pheromone update) speed and decreasing digital mark value (pheromone evaporation) speed; input data dynamic changes frequency – IF.

So Smax could be represented as next dependency:

$$\mathrm{SMAX} = \mathrm{FUNCTION}(N, \mathrm{VIJ}, \mathrm{IF}). \tag{4}$$

In accordance with expression (3) digital mark value Sij now is limited by a range between 0 and Smax.

Proposed approach with adaptive top limit for digital mark values enables to limit mark values on links between nodes. It enables to search alternative (more optimal) routes by ants-agents even when dynamic input data was not changed for a long time. And even when input data was changed after long-term stability, developed computer system can adapt to new conditions, which was impossible before.

CONCLUSIONS

Local optimization methods implementation into developed computer system for solving TSP with swarm agent's behavior model enables to reduce difference between quasi-optimal and optimal routes and reduce iteration count of routes search by ants-agents cycle. Developed computer system demonstrates acceptable in practice results in calculated time and in accuracy of computations. Developed computer system in experiments with tested TSP (532 nodes) in dynamic input data conditions could achieve quasi-optimal resulting route with difference from optimal less than 6 % in less than 1 second of calculation. Also, in accordance with TSPs solving results developed computer system can find quasi-optimal route with difference from optimal result less than 3 % for TSPs with size less than 6000 nodes.

Routing problem in real-life computer network can be represented as a dynamic asymmetrical TSP (DATSP) with a partly unknown input data. During such type of TSP solving the following problems appear: possibility for agents to get into "trap" or "deadlock" situations; impossibility of classic digital mark values update; inefficient found resulting routes reusing; infinite increase of digital mark values and, as a result, endless "ant colony memory" strengthening.

Developed overcoming and processing "trap" and "deadlock" situations algorithms enable ants-agents to continue solving TSP in case of such critical situations. Proposed new approaches: digital mark values increase (”pheromone update”), after each link between nodes was passed, and approach with adaptive top digital mark values limit Smax enables to use placed on nodes digital mark values correctly and to adapt for new conditions quickly when dynamic input data changes have place.

Also, an approach to the best routes calculation and data transfer processes union, during which search and analysis for optimal routes process is united with practical task – data transfer inside the network was proposed. It means that during data transfer by ants-agents, search for best routes (minimal by cost with aim to decrease expenses) logic is executed. In scope of such approach for developed computer system, found routes are actual in the moment of their calculation.

REFERENCES

[1] Cormen, T., Leiserson, C., Rivest, R., Stein, C. Introduction to Algorithms, third edition. MIT Press, 2009, p. 1292.

[2] Bonabeau, E., Dorigo, M., Theraulaz, G.. Swarm intelligence: From Natural to Artificial Systems. Oxford University Press., 1999, p. 307.

[3] Muliarevych O., Golembo V. New approaches for solving travelling salesman problems using agents swarm intelligence behavior model, // European Cooperation: Scientific Approaches

and Applied Technologies. – CLM Consulting Publ., Warsaw 2015, Vol 5(5), pp. 131–143.

[4] Lentin Joseph, Mastering ROS for Robotics Programming. Packt Publishing Ltd, UK, Dec 21, 2015, p. 480.

[5] Golembo, V., Muliarevych, O.. Modification of ant colony method for TSP solving by collective of autonomous agents. Computer systems and networks, Lviv Polytechnic National University Press, no. 717, 2011, pp. 24–30.

[6] E. N. Barron, Game Theory: An Introduction. John Wiley & Sons Publ., Canada, Apr 22, 2013, p. 555.

[7] Helsgaun, K., General k-opt submoves for the Lin-Kernighan TSP heuristic. Mathematical Programming Computation, Springer Verlag Publ., 2009, pp. 119–163.

[8] Heidelberg University (from 1995). Discrete and Combinatorial Optimization. Retrieved from http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

Oleksandr Muliarevych received the B. S. degree in computer engineering from Lviv Polytechnic National University, Ukraine, in 2010 and the M. S. degree in computer system and networks from Lviv Polytechnic National University, Ukraine, in 2012. In 2016 pursued the Ph. D degree in computer system and components in Lviv Polytechnic National University, Ukraine.

From 2014 he is an Assistant of the Computer Engineering Department in Lviv Polytechnic National University. His research interest includes the design of multi-agent system, “smart home” based computer systems, “IoT” type devices and solution development, calculation optimization and methods, utile and systems development in logistic.