

Parallel Ant Colony Algorithm for Shortest Path Problem

Géza Katona^{1*}, Balázs Lénárt², János Juhász³

¹ Department of Automotive Technologies

Faculty of Transport Engineering and Vehicle Engineering (KJK),
Budapest University of Technology and Economics, H-1111 Budapest Stoczek u. 6., Hungary

² Department of Material Handling and Logistics Systems,

Faculty of Transportation Engineering and Vehicle Engineering (KJK),
Budapest University of Technology and Economics, H-1111, Budapest Bertalan L. u. 7–9., Hungary

³ Department of Transport Technology and Economics,

Faculty of Transportation Engineering and Vehicle Engineering (KJK),
Budapest University of Technology and Economics, H-1111, Budapest Stoczek u. 2., Hungary

* Corresponding author, e-mail: geza.katona@gjt.bme.hu

Received: 06 July 2018, Accepted: 14 November 2018, Published online: 28 January 2019

Abstract

During travelling, more and more information must be taken into account, and travelers have to make several complex decisions. In order to support these decisions, IT solutions are unavoidable, and as the computational demand is constantly growing, the examination of state-of-the-art methodologies is necessary. In our research, a parallelized Ant Colony algorithm was investigated, and a parameter study on a real network has been made. The aim was to inspect the sensibility of the method and to demonstrate its applicability in a multi-threaded system (e.g. Cloud-based systems). Based on the research, increased effectiveness can be reached by using more threads. The novelty of the paper is the usage of the processors' parallel computing capability for routing with the Ant Colony algorithm.

Keywords

shortest path, parallel computing, multimodal transportation network, Ant Colony algorithm, super network, cloud computing

1 Introduction

The multimodal network should contain a certain number of data that the algorithms have to process in the shortest possible time. This paper aims to lay down the foundation for our further work in a large-sized multimodal network.

In our previous research paper, the weaknesses of traditional algorithms were introduced [1]. Their main drawback is that they produce only one solution per one parameter set and they do not give alternative solutions (multiple routes). Using an artificial intelligence method - in our case, Ant Colony algorithm - is a possible way to calculate multiple alternative options; moreover, this system is capable of taking multiple input parameters (criteria) into account. The drawback is: Ant Colony is a resource-intensive method, however, the parallelization of the method could be expedient on large networks.

More and more researchers are involved in this field, as time saving, better efficiency and cost optimizations are getting more important. In 2012, a paper deeply analyzing this topic was published at the Eindhoven University [2].

They introduced the multimodal routing system. In this paper, the Dijkstra [3] [4] and A* [4] [5] algorithms were used. The conclusion was that there is some limitation in large networks. To solve this problem, it is possible to use caching techniques or to limit the search area, which was introduced in a paper where a super network was used. [6]

Dominik Kirscher used the Ile-de-France region in France in his research [7], and his aim was to find a route with the same origin and destination. This system is perfect for commuters who use P+R facilities every day, and go in the city center with public transport; and, at the end of the day, they arrive back to the same P+R place. In this research, the SDALT algorithm was used instead of Dijkstra to speed up the calculations.

The old traveling salesman problem (TSP) and its solutions shall not be forgotten either. One of the solutions is the Branch and Bound Algorithm. G. Laporte and Y. Nobert made research with this method [8]. Their aim was to apply the algorithm on capacitated vehicles with high number of

cities. This development was continued in 2003 by Jens Lysgaard, Adam N. Letchford and Richard W. Eglese [9]. The original method was supplemented by cut pool management to increase its efficiency. Some years later, the robustness was increased [10].

A genetic algorithm based solution was implemented by Haicong Yu and Feng Lu [11]. Their system, with an improved genetic algorithm was capable of handling a multimodal network. They showed that the genetic algorithm has the ability to investigate the options simultaneously.

A Helsinki commuter's network was investigated by 3 scientists [12]. An application was developed where the public transport and P+R facilities were linked together. As a result, the level of CO₂ emission was decreased and P+R usage was increased.

A Hungarian survey was made to compare the available route planners in Hungary [13]. This paper summarized the abilities of the different planners' services, focusing on multimodality.

Moreover, not just the algorithm, but the hardware is also important. In the first period of modern computer technology, transistor miniaturization was the solution to increase the computing capability. This was the basis of Moore's law which was originally a prediction by Gordon Moore [14]. He was the co-founder of Fairchild Semiconductor Inc. and Intel corporations. In the last decades of the 20th century, computer technology started to reach an upper limit of the possible computing speed, what was that time the clock frequency of the transistors, expressed in Hz. Therefore, researchers were looking for another possible option. The solution was the ability of computing simultaneously, also called multi-threading [15].

The Ant Colony algorithm is widely used and researched in the Civil Engineering area. The algorithm was applied to structural calculation by Anikó Csébfalvi [16] [17] [18] [19] [20]. In these papers, the Ant Colony algorithm (ACO) was combined with the Genetic algorithm (GA) and the local search (LS) solution. The method called Angel was based on the evaluation of the approaches ACO, GA and LS. This complex approach resulted in significant time saving, and generated a more stable result. With this solution, the elastic-plastic material load-deflection can be handled. Moreover, a research was made to compare the capability of hybrid metaheuristic methods [21] and also hyperheuristic methods [22]. A special usage of these methods was a project scheduling case to support a water management program [23]. A deeper research of the Angel algorithm leads to a new approach by Anikó Csébfalvi [24] [25]. This

possible combination was also researched by Ali Kaveh and Siamak Talatahari [26]. In their article, the heuristic particle swarm ant colony optimization (HPSACO) was introduced. Thanks to the development, the improved algorithm was more efficient and more robust; therefore, higher convergence can be reached than with former solutions. The aim of the research was to achieve a more effective tool to support the calculation for the optimal design of trusses. Another aim was the development of the design space's extensions. In this approach, a partitioning technique is required to avoid the fall of the searching speed [27].

A parallel computing technique was examined by Péter Iványi [28, 29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44]. The aim was to use the modern multi-threaded processors' capability to increase the computational speed. The research scope was parallel computing, later on cloud computing; and, last but not least, CUDA (Compute Unified Device Architecture) programming. CUDA is a parallel computing interface developed by Nvidia [45]. With this technique, the GPU's (Graphics Processing Unit) more effective parallel computing capability can be used instead, or, in this case, it can supplement the less effective CPU (Central Process Unit). This technique is also used in Discrete Element Method (DEM) [46]. The parallel and asynchronous grid computation methods are also researched in connection with DEM for American options derivatives [47] and for the simulation of 3D electrophoresis [48].

The routing problem usually needs to take multiple criteria into account. There are plenty of solutions conceivable in addition to the methods listed above. Heinz Spiess and Michael Florian [69] described a possible method to formulate the transit assignment problem to the linear programming problem. Moreover, non-linear cost extension is also taken into account.

The algorithm for real time network was developed by Jerald Jariyasunant, Eric Mai and Raja Sengupta [49]. Here, the main technique was the combination of a pre-calculated network and the real dataset. With the pre-calculation, sufficient computing time could be reached.

A routing program prototype was made by Man-chun Tan, C.O. Tong, S.C. Wong and Jian-min Xu in Hong Kong and Guangzhou area [50]. The main expectations were acceptable time and the transfer-walk criterion during the routing process.

A modular transportation approach was presented by Péter Bocz, Lajos Kisgyörgy and Gergely Vasvári [51]. In this paper, a demand-controlled system with a statistical impact was described.

Another approach is stochastic traffic assignment [52]. The imitation of the transport system was the aim with the Stochastic User Equilibrium (SUE) algorithm. With this technique, the condition and performance of the network can be rapidly determined. The logit model [53] was also considered.

In a wider scope, there are plenty other algorithms in the Civil Engineering area that can also make up a good basis for routing. A good approach could be Enhanced Colliding Bodies Optimization (ECBO), which is used for size optimization in the case of monopole steel structures [54]. Another research scope is the Volume-constrained Expected Compliance Minimization with Probabilistic Loading Directions supported by the optimal solution searching optimality criteria (OC) method [55]. An efficient approach is the shortest route trees, that helps the meshing procedure [56].

Calculation time is a key factor during the routing process, especially in a multimodal case, which results in a more complex network. Moreover, it is also worth mentioning Y. Sheffi's work [6] with the extension of the autonomous vehicles [57]. The smart city [58] conception should also be taken into account, which assumes a higher level of cooperation (IoT¹) or control.

One of the main focuses of autonomous car development is to implement more and more effective algorithms for routing. This is based on the fact, that the main computer needs to process plenty of sensor signals to create a special model for the driving algorithm. This process requires enormous computation capacity which causes high heat emission and requires large space. On one hand, this issue can be handled by the processor manufacturers [59] [60] [61] [62] [63]; however, an accurate and fast algorithm is also needed, which can use the parallel computing capability of these processors.

Based on our previous research and the studies mentioned above, the hypothesis is the following: if the iteration number of a heuristic route planning algorithm (in this case the Ant Colony) is decreased and the parallel computing option is used instead of the iteration, the total time usage decreases. In addition, the resource utilization increases. In this study, an ant corresponds to a processor's thread. The innovation of the paper is the parallel computational Ant Colony algorithm and the adequate parameter setting for parallel computation based on test results.

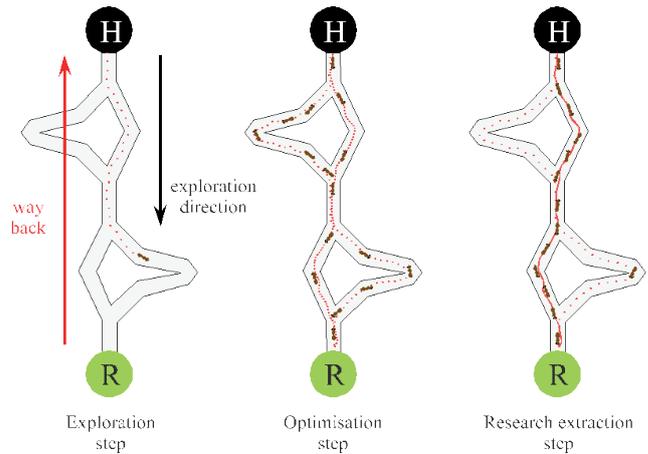


Fig. 1 The operation of the Ant Colony algorithm

2 Methodology – Implementation of the Ant Colony algorithm

The Ant Colony algorithm was invented by Marco Dorigo [64] [65]. This method simulates an ant colony's resource gathering process. The aim of the hive (colony) is to find enough resource located in attainable distance. The ants developed a method to serve the effective operation of the hive. In every case, an explorer ant is sent. If the ant finds something needed by the hive, the route is signed with pheromone vapor on the way back. The next ant is able to follow this sign or choose another direction. The stronger the pheromone sign, the more ants follow. It is easily understandable that closer or bigger the resource is, the stronger the sign is. If the path is used less frequently, the pheromone sign evaporates. The following figure helps to understand this method (Fig. 1), where H is the hive and R is the resource.

One of our previous papers [1] investigated the related research works in connection with this field, and laid down the methodological basis of the current work. The model is based on choosing the probability of an actual connection [66]. The essence of the method is the pheromone change that indicates the level of suitability of the path.

$$\Delta\tau_{ijk} = \frac{Q}{d_k} \quad (1)$$

In Eq. (1), $\Delta\tau_{ijk}$ indicates the pheromone change, $[d_k] = (m)$ is a route length generated by the actual ants, and $[Q] = (m)$ is a factor to control the importance of the distance. i and j are identifiers of the node. k is the identifier of the ant.

The next step is to collect the pheromone changes from the individual ants.

$$\Sigma \Delta \tau_{ij} = \Sigma \Delta \tau_{ij} + \Sigma \tau_{ijk} \quad (2)$$

$\Sigma \Delta \tau_{ij}$ is the summarized pheromone change between i and j . Thus, the following equation describes the update process:

$$\tau_{ij} = \tau_{ij} \cdot (1 - \rho) + \Sigma \Delta \tau_{ij} \quad (3)$$

This equation summarizes the ants' path information and the evaporation of the pheromone. Here, $[\rho] = (-)$ is the evaporation factor and $[\tau_{ij}] = (-)$ is the actual quantity of the pheromone.

Based on the previous equation, the choosing probability can be calculated as

$$p_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{u \notin M_k} (\tau_{iu})^\alpha (\eta_{iu})^\beta} & \text{if } j \notin M_k \\ 0 & \text{if } j \in M_k \end{cases} \quad (4)$$

where $[\eta_{ij}] = (1/m)$ is the inverse of the measured length between i and j , $[\alpha] = (-)$ is the weight of the pheromone, $[\beta] = (-)$ is the weight of the distance and M_k is the route of the specified ant.

At the beginning, the network is unknown, and it is discovered by the ant movements. By increasing the efficiency of the convergence, an elite ant is chosen. This feature is shown in the following figure, where the structogram [67] representation technique was used (Fig. 2).

Some protections are implemented to the algorithm, which help to avoid backtracking (dead end) and loops.

To make sure that the algorithm finds a solution, another feature was built in. If an ant is stuck in a dead end, it steps back to the last junction. This procedure is shown below (Fig. 3, Fig. 4, Fig. 5)

A dead end black list was made, in which the nodes with no further connections can be stored. If a node is stored in this list, the possibility of choosing that is changed to zero. This way, a sufficient solution and acceptable calculation time can be reached. If the destination is unreachable, an infinite distance is given. In 1965, two scientists presented a similar algorithm [68]. The result of the first case (Fig. 3, Table 1), the second case (Fig. 4, Table 2) and the third case (Fig. 5, Table 3) can be found below.

$length_{min} > length_{actual}$		/
$length_{min} = length_{actual}$		
$route_{best} = route_{actual}$ $best_{ant} = ant_{actual}$		

Fig. 2 Structogram for elitism

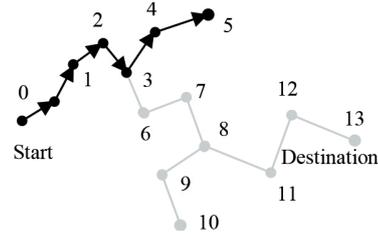


Fig. 3 Example network – First case: dead end was found

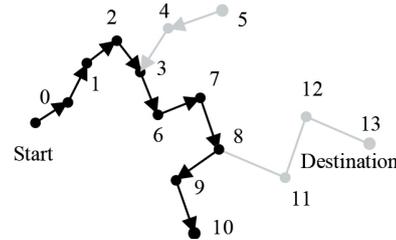


Fig. 4 Example network – Second case: dead end was found

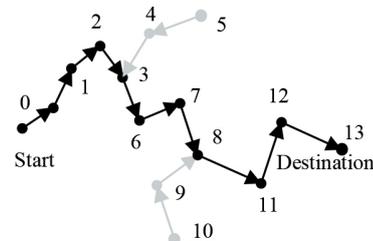


Fig. 5 Example network – Third case: destination was found

Table 1 The result of the first case

Route	0, 1, 2, 3, 4, 5
Dead end list	-

Table 2 The result of the second case

Route	0, 1, 2, 3, 6, 7, 8, 9, 10
Dead end list	4, 5

Table 3 The result of the third case

Route	0, 1, 2, 3, 6, 7, 8, 11, 12, 13
Dead end list	4, 5, 9, 10

The aim and also the novelty in this paper is speeding up the algorithm with parallel computing. This idea is based on nature and also on modern computers. There are several ants in nature which explore the environment around the hive. This can be simulated with multi-threaded systems. This article is in connection with a previously issued conference paper. [1].

The overview of the multi-threaded code is shown in Fig. 6, where each thread belongs to an Ant. The network is discovered by the ants, and the route attributes ($\Delta \tau$) are shared with each other after they return back to the hive.

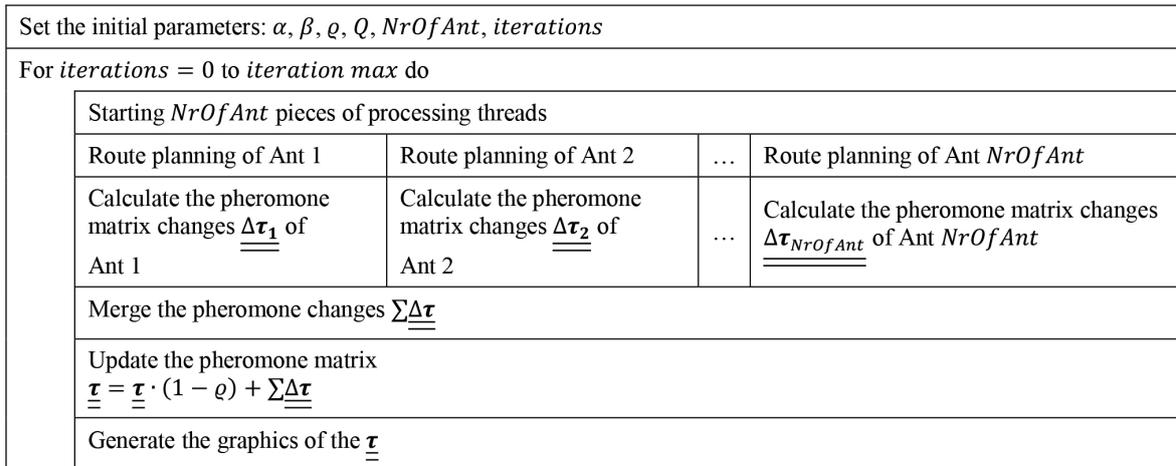


Fig. 6 Structogram of the parallel Ant Colony algorithm

The pheromone matrix is updated after each iteration. The result of this method is a decreased number of iterations and time saving with the parallel computation, where each thread belongs to an Ant.

3 Results

3.1 Application on test network

The database implements the structure of the GTFS [General Transit Feed Specification]. There are public transport routes from Budapest and from Vienna. Some of the possible connections are stored, e.g.: railroads, bus and air lines. These are listed in Table 4 and shown on a network map (Fig. 7).

The GTFS data contains valid data from different transport operators and the links are shown in Fig. 7.

3.2 Initialization of the test

As outlined by the data, the algorithm is influenced by several factors that can modify the result. Therefore, a parameter study was implemented to examine the connection between the results and the factors. A HP ProLaint ML310e Gen8 v2 was used for the tests. The computer has an Intel Xeon E3-1220 CPU which runs on 3.10 GHz, and has 4 GB of system memory. The application was written in Java which runs on Ubuntu 14.04.4 LTS with OpenJDK Runtime Environment version 1.7.0_95. The following parameters were reviewed: $\alpha, \beta, \rho, Q, NrOfAnt, Threads, iterations$ (Table 5). Parameters α, β, ρ, Q modify the operation of the algorithm, and the other two affect the efficiency.

Furthermore, the execution time of the algorithm was also registered. All ants were supervised in every execution and in each iteration from the point it has been created

Table 4 Transport trips in GTFS

Destination	Route
M1	Mexikói út M / Vörösmarty tér
M2	Örs vezér tere M+H / Déli pályaudvar M
M3	Kőbánya-Kispest M / Újpest-központ M
4	Újbuda-központ / Széll Kálmán tér M
6	Móricz Zsigmond körtér / Széll Kálmán tér M
7	Bosnyák tér / Albertfalva vasútállomás
173	Bornemissza tér vá. / Újpalota, Nyírpalota út
200E	Határ út M / Liszt Ferenc Airport 2
Railjet	Budapest-Keleti / Wien Westbahnhof
912/942	Budapest-Keleti / Sopron
2821	Sopron / Wien Meidling
130	Budapest-Keleti / Bratislava Hlavná Stanica
2513	Bratislava Hlavná Stanica / Wien Hauptbahnhof
U1	Reumann Platz / Leopoldau
U2	Karlsplatz / Aspernstraße
U3	Ottakring / Simmering
U4	Hütteldorf / Heiligenstadt
U6	Siebnhirten / Floridsdorf
CAT	Bahnhof Wien Mitte / Flughafen Wien
S7	Rennweg / Flughafen Wien
Volán	Budapest, Népliget aut. pu. / Wien, VIB, U3 Station Erdberg
OS 714	BUD Nemzetközi repülőtér / Flughafen Wien

until it reached the defined target. Additionally, the time of the iteration was also registered. This data will be important in the future, when parallel operation will be reviewed. A TOP 5 list was created from the results of the shortest route length. The number of occurrences was counted to each of the lengths in every case. The parameters were evaluated on the basis of the number of occurrences.

First, parameters *NrOfAnt* and *Threads* were reviewed (Fig. 8).

Table 5 Values of parameter study

Parameter	Explanation of the parameter	Value
α	Pheromone importance indicator	0.1; 0.6; 1.1; ... 5.1
β	Distance importance indicator	0.05; 0.15; ... 0.45
ρ	Control the loss of pheromone	0.1; 0.2; 0.3; ... 0.9
Q	Control the distance	100; 600; ... 5100
NrOfAnt	Number of ants	1; 2; ... 8
Threads	Number of threads	1; 2; ... 8
iterations	Number of iterations	10

The points (Fig. 8) were the result of the number of ants multiplied by the measured distance. On the X-axis, the number of ants and the number of threads used are shown, and the Y-axis represents the calculated points. Where the number of threads is smaller than the number of ants, the ants wait until a thread becomes free. Every *NrOfAnt* and *Threads* settings are executed with the same α , β , ρ , Q parameters. After a closer look at the results, it can be declared that using only 1 thread results in the highest points until 6 ants. Consequently, the higher number of threads has an effect over 7 ants. Using these settings for the algorithm, the result is stably TOP 1 in approx. 40% of the cases from the sum of the TOP 5 results, which is a good outcome.

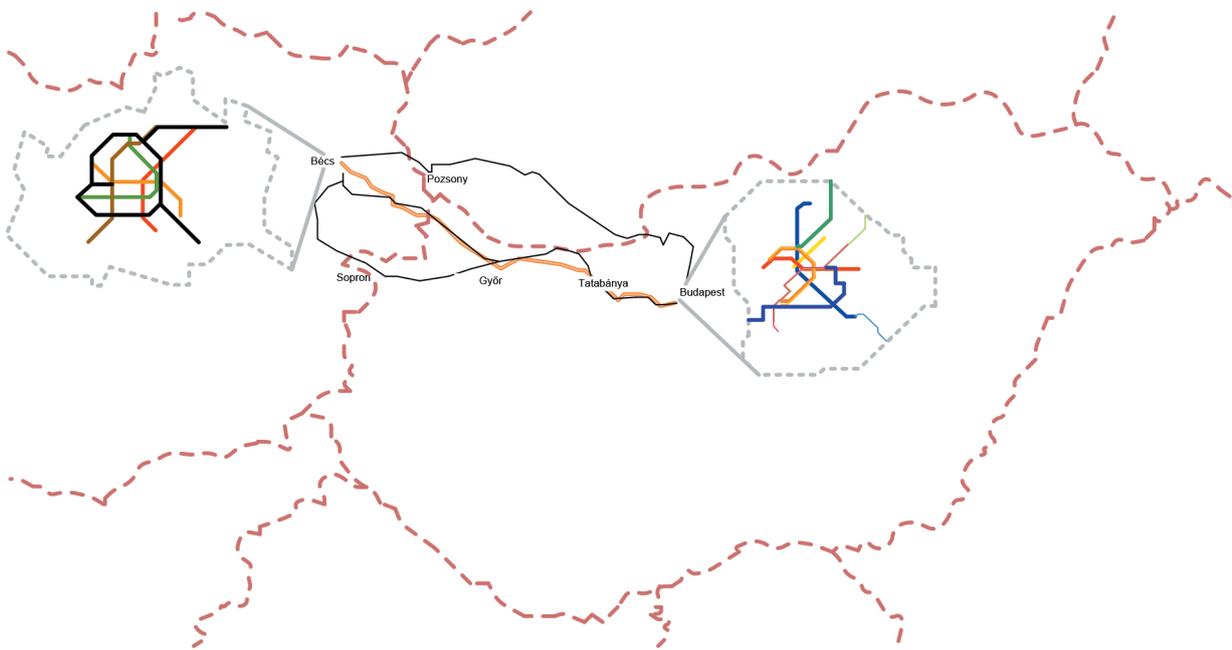


Fig. 7 Transport network

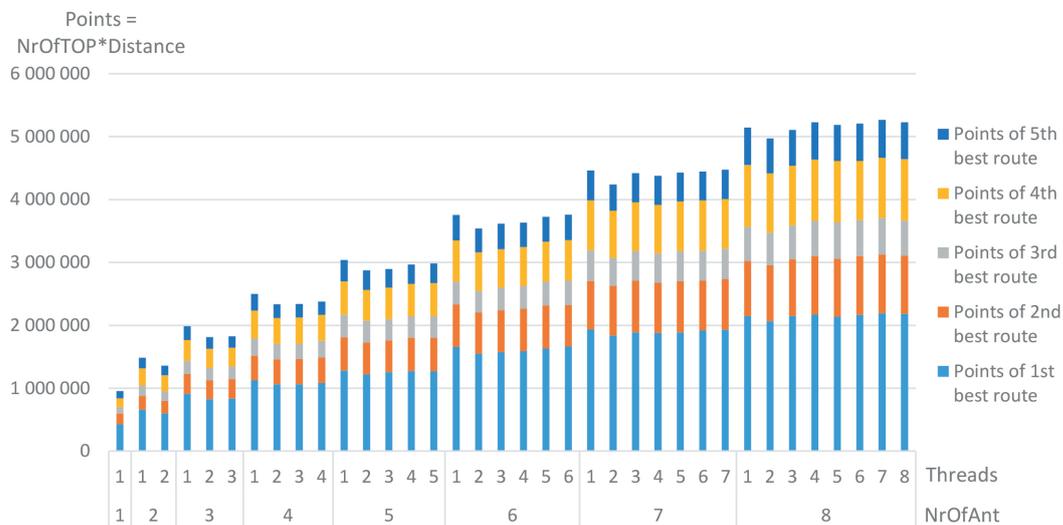


Fig. 8 Parameters *NrOfAnt* and *Threads*

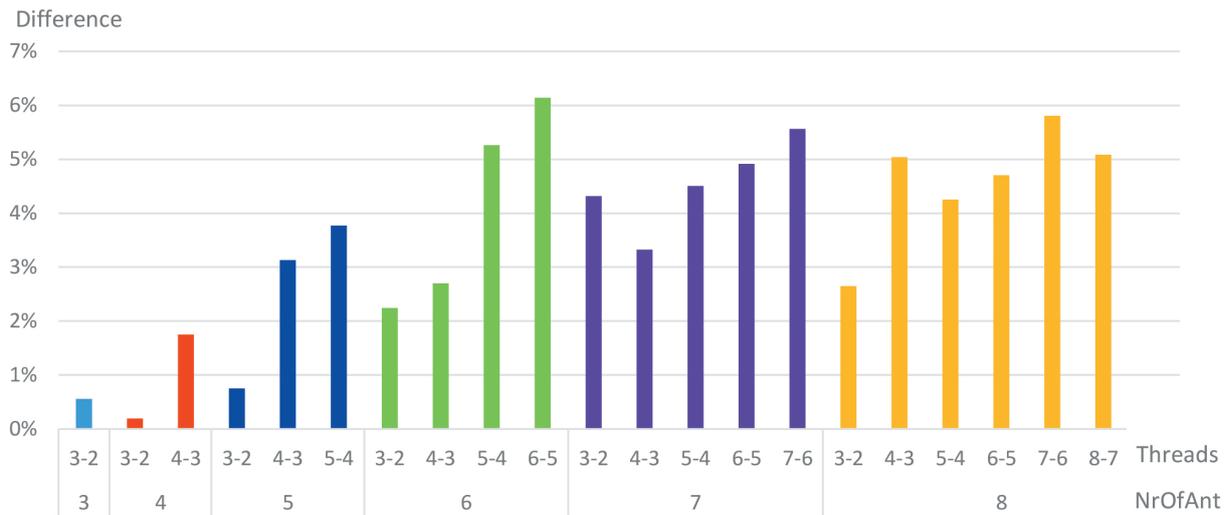


Fig. 9 Cumulated Ratio of point differences

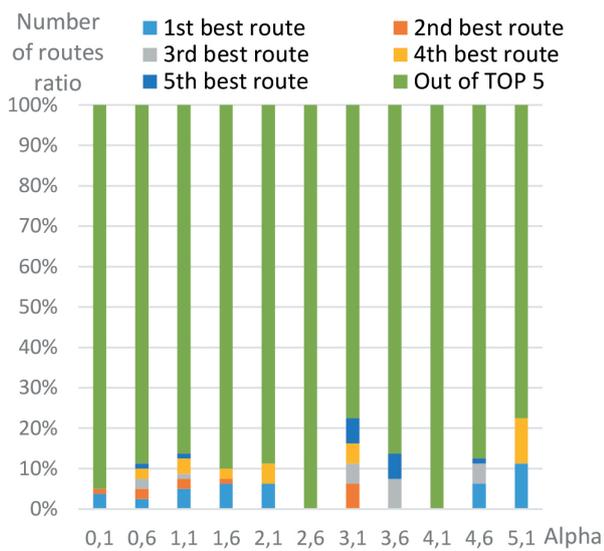


Fig. 10 The result of $\alpha(\beta = 0.05; Q = 100; \varrho = 0.1)$

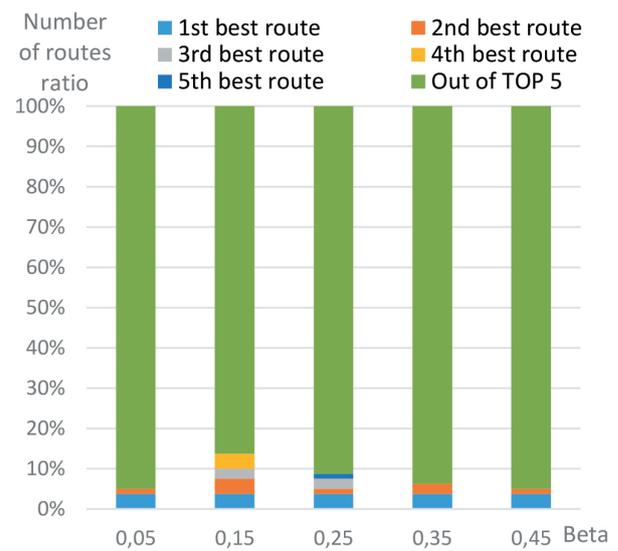


Fig. 11 The result of $\beta(\alpha = 0.05; Q = 100; \varrho = 0.1)$

Continuing the data analysis, the steps are also important between the different settings. The results are presented in Fig. 9.

The diagram indicates the cumulated ratio differences in a similar *NrOfAnt* group. Based on this, it is worth raising the number of threads, because the cumulated thread differences are always positive. And at *NrOfAnt* = 8, the result is not just relatively positive, but it is absolutely positive, too (Fig. 8).

Consequently, in the next step with *NrOfAnt* = 8 and *Threads* = 8 settings, the α , β , ϱ , Q parameters are investigated. In the following figure (Fig. 10), the α parameter results are presented. The horizontal axis shows the alpha value and the vertical axis represents the number of the

routes ratio. So for example the $\alpha = 0.1$ settings eventuate approx. 4% TOP 1 and 1% TOP 2 routes. So, based on this, parameters $\alpha = 3.1$ and $\alpha = 5.1$ cause the highest number of TOP routes.

Fig. 11 presents the β parameter results. Based on my previous investigation, β is set between 0.05 and 0.45, because it has not been effective over 0.5. [1] Based on this result, β under 0.25 is effective.

The result of ϱ is presented in Fig. 12. The highest ratio of the best route can be found between 0.5 and 0.9. Raising the value of ϱ is causing higher number of the TOP 5 result.

Last, but not least, results for Q are in the figure above (Fig. 13). The diagram shows that Q over 1600 causes more TOP results.

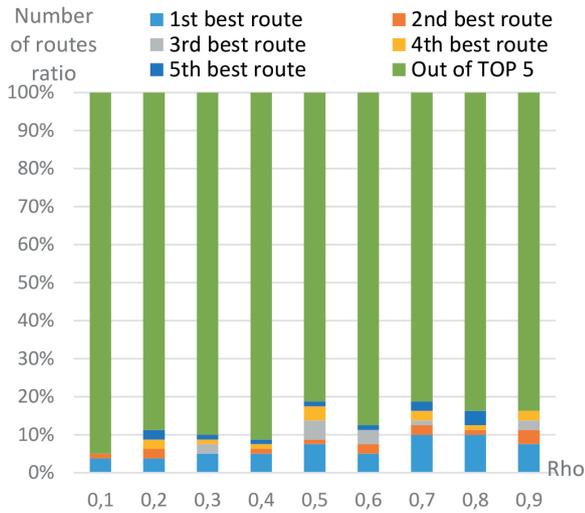


Fig. 12 The result of $Q(\alpha = 0.05; \beta = 100; Q = 0.1)$

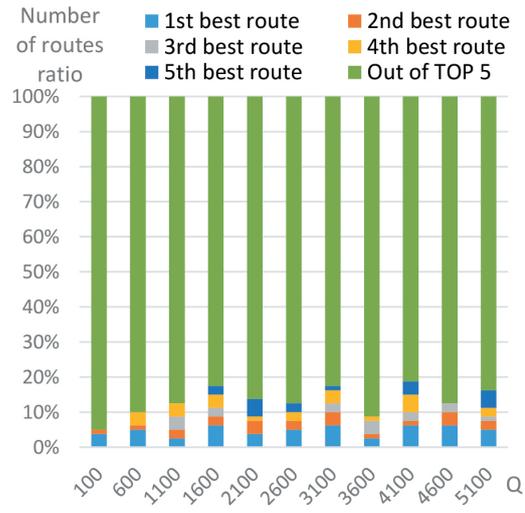


Fig. 13 The result of $Q(\alpha = 0.05; \beta = 100; \varrho = 0.1)$

4 Analysis and discussion

Each result is calculated after 10 iterations. The pheromone matrix (τ) was updated after every step. This can be graphically seen in the following pictures (Fig. 14). One can see 9 examples from the 10 iterations below.

In the figure, darker pixels mean stronger pheromone evaporation rates. The vertical axis represents the source and the horizontal axis represents the destination. The structure explanation of the matrix can be found in Fig. 15.



Fig. 14 The changes of the pheromone matrix

	1. column	...	<i>j.</i> column	...	<i>n.</i> column
1. row					
⋮					
<i>i.</i> row			<i>i., j.</i> pixel		
⋮					
<i>n.</i> row					

Fig. 15 The graphics of the pheromone matrix structure

After the parameters were investigated, the overall outcome was tested. The goal is to find alternative routes along the best route. The result was summarized and filtered. Those parameters were collected by which all TOP 5 routes were found. Efficiency was calculated between all possibilities and the TOP 5 routes. A top list was made from the efficiency (Table 6), this list contains values over 20%.

Examining the data (Table 6) shows that only the value of β is fixed, the other values are less stable.

In the next step, the execution time was investigated (Fig. 16)

In the diagram, the average individual ant time and the average sum time is demonstrated. In the second y axis the ratio of the average sum and the average individual time was presented. The decrease in the ratio means that the sum and the individual average time are getting closer. Consequently, the meaning of the ratio is that the time gain is getting smaller with a higher number of ants.

5 Conclusions

In this study, the parallelization of the Ant Colony algorithm was introduced based on previous research results, for example in Civil Engineering area. Different parameters and thread counts were investigated. In conclusion, we can emphasize that it is worth using more threads, because the probability of finding valid routes is increasing. In addition, time usage and resource utilization are also better. This capability is beneficial in a large-sized network. Based on the results, the hypothesis is considered to be supported.

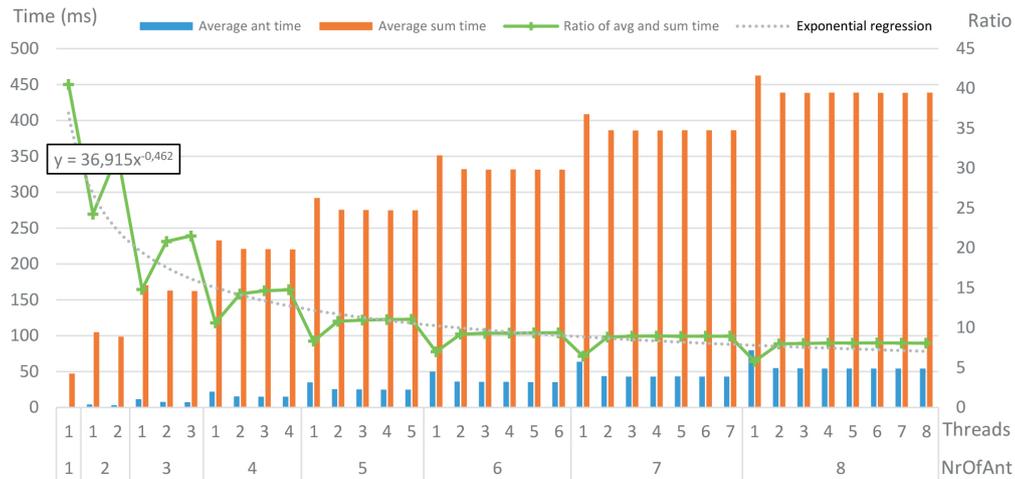


Fig. 16 Run time

Table 6 The best parameters of the Ant Colony algorithm
 (Iteration = 10, NrOfAnt = 8 and Threads = 8)

α	β	ρ	Q	1 st best route	2 nd best route	3 rd best route	4 th best route	5 th best route	SUM of TOP 5	Sum of all routes	Efficiency
3.1	0.05	0.3	600	9	5	3	7	1	25	80	31%
0.1	0.05	0.4	4100	9	4	2	1	3	19	80	24%
1.1	0.05	0.8	1600	9	3	2	1	1	16	80	20%
1.1	0.05	0.4	4100	8	3	2	2	1	16	80	20%
0.6	0.05	0.5	100	8	1	1	2	4	16	80	20%
4.1	0.05	0.2	1600	7	3	2	2	3	17	80	21%
1.1	0.05	0.7	1100	7	3	1	3	3	17	80	21%
5.1	0.05	0.6	1100	7	1	4	5	1	18	80	23%
1.1	0.05	0.5	2100	6	4	2	2	2	16	80	20%
0.6	0.05	0.6	5100	6	3	4	2	4	19	80	24%
1.1	0.05	0.9	2600	6	3	3	5	3	20	80	25%
0.6	0.05	0.2	2600	6	3	3	4	1	17	80	21%
1.6	0.05	0.8	1600	6	1	4	3	3	17	80	21%
1.1	0.05	0.6	2600	6	1	4	2	3	16	80	20%
2.1	0.05	0.6	5100	5	4	2	5	1	17	80	21%
0.6	0.05	0.4	100	5	4	2	2	5	18	80	23%
1.1	0.05	0.4	4600	5	3	1	4	4	17	80	21%
2.6	0.05	0.1	4100	5	2	4	3	2	16	80	20%
0.1	0.05	0.7	3600	5	1	3	3	5	17	80	21%
0.6	0.05	0.9	3100	4	6	3	3	1	17	80	21%
1.6	0.15	0.2	600	4	4	4	5	2	19	80	24%
5.1	0.05	0.7	2600	4	4	3	4	3	18	80	23%
0.6	0.05	0.6	3600	4	4	2	3	4	17	80	21%
0.6	0.05	0.2	2100	4	3	4	3	2	16	80	20%
1.6	0.05	0.4	5100	3	4	5	3	2	17	80	21%
2.1	0.05	0.8	3600	2	3	4	4	5	18	80	23%

In further research, this algorithm will be applied for large-sized networks on a multi-core system, where more threads will be available. Further development will be the determination of the full resilience function, which contains not just the distance used in this paper, but the travelling time and some other factors, too. Another research aim is to integrate autonomous vehicles in the routing process. This can be one supporting element for the Smart City conception and a central traffic management system.

References

- [1] Katona, G., Lénárt, B., Juhász, J. "Compare Ant-colony and Genetic algorithm for shortest path problem and introduce their parallel implementations", In: 4th International Conference on Models and Technologies for Intelligent Transportation Systems, Budapest, Hungary, 2015, pp. 312–319. <https://doi.org/10.1109/MTITS.2015.7223273>
- [2] Zang, J., Liao, F., Arentze, T., Timmermans, H. "A Multimodal Transport Network Model for Advanced Traveler Information System", Procedia – Social and Behavioral Sciences, 20(4), pp. 313–322. 2011. <https://doi.org/10.1016/j.sbspro.2011.08.037>
- [3] Dijkstra, E. W. "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, 1(1), pp. 269–271, 1959.
- [4] Podobni, K. "Legrövidebb útkereső algoritmusok", BSc Thesis, Eötvös Loránd University, 2009. Available at: https://web.cs.elte.hu/blobs/diplomamunkak/bsc_matelem/2009/podobni_katalin.pdf [Accessed: 08 January 2019]
- [5] Hernáth, Z. "Real-time adaptive A* routeplanning algorithm", Budapest University of Technology and Economics, Budapest, 2012. (in Hungarian: "Valós idejű adaptív A* útkeresési algoritmus")
- [6] Sheffi, Y. "Urban transportation networks: Equilibrium Analysis with Mathematical Programming Methods", Pentice-Hall Inc., Englewood Cliffs, New Jersey, USA, 1985.
- [7] Kirchler, D. "Efficient routing on multi-modal transportation networks", Thesis, Ecole Polytechnique X, Palaiseau, France, 2013.

- [8] Laporte, G., Nobert, Y. "A branch and bound algorithm for the capacitated vehicle routing problem", *Operations-Research-Spektrum*, 5(2), pp. 77–85, 1983.
<https://doi.org/10.1007/BF01720015>
- [9] Lysgaard, J., Letchford, A. N., Eglese, R. W. "A new branch-and-cut algorithm for the capacitated vehicle routing problem", *Mathematical Programming*, 100(2), pp. 423–445, 2004.
<https://doi.org/10.1007/s10107-003-0481-8>
- [10] Fukasawa, R., Longo, H., Lysgaard, J., de Aragão, M. P., Reis, M., Uchoa, E., Werneck, R. F. "Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem", *Mathematical Programming*, 106(3), pp. 491–511, 2006.
<https://doi.org/10.1007/s10107-005-0644-x>
- [11] Yu, H., Lu, F. "A multi-modal route planning approach with an improved genetic algorithm", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(II), pp. 344–348, 2012.
- [12] Noreikis, M., Butkus, P., Nurminen, J. K. "In-Vehicle Application for Multimodal Route Planning and Analysis", In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, Luxembourg, 2014. pp. 350–355.
<https://doi.org/10.1109/CloudNet.2014.6969020>
- [13] Esztergár-Kiss, D., Csiszár, Cs. "Multicriteria Analysis of Hungarian Journey Planners", *Periodica Polytechnica Transportation Engineering*, 44(2), pp. 97–104, 2016.
<https://doi.org/10.3311/PPtr.8570>
- [14] Moore, G. E. "Cramming more Components onto Integrated Circuits", *Proceedings of the IEEE*, 86(1), pp. 82–85, 1965.
<https://doi.org/10.1109/jproc.1998.658762>
- [15] Iannucci, R. A. "Toward a dataflow/von Neumann hybrid architecture", In: ISCA '88 Proceedings of the 15th Annual International Symposium on Computer architecture, Honolulu, Hawaii, USA, 1988, pp. 131–140.
<https://doi.org/10.1145/633625.52416>
- [16] Csébfalvi, A. "Robust Truss Optimization with Uncertain Load Directions", In: Proceedings of the Third International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Stirling, Civil-Comp Press, Stirlingshire, UK, 2013, pp. 1–10.
<https://doi.org/10.4203/ccp.103.11>
- [17] Csébfalvi, A. "Angel method for discrete optimization problems", *Periodica Polytechnica Civil Engineering*, 51(2), pp. 37–46, 2007.
<https://doi.org/10.3311/pp.ci.2007-2.06>
- [18] Csébfalvi, A. "A hybrid meta-heuristic method for continuous engineering optimization", *Periodica Polytechnica Civil Engineering*, 53(2), pp. 93–100, 2009.
<https://doi.org/10.3311/pp.ci.2009-2.05>
- [19] Csébfalvi, A. "ANGEL: A simplified hybrid metaheuristic for structural optimization", In: Helio, J. C. (ed). *Ant Colony Optimization*, 1st ed., IntechOpen, London, UK, 2013, pp. 107–127.
<http://dx.doi.org/10.5772/52188>
- [20] Csébfalvi, A. "Multiple constrained sizing-shaping truss-optimization using ANGEL method", *Periodica Polytechnica Civil Engineering*, 55(1), pp. 81–86, 2011.
<https://doi.org/10.3311/pp.ci.2011-1.10>
- [21] Csébfalvi, A. "Hybrid Metaheuristic Methods in Truss Optimization: A Review", *Computational Technology Reviews*, 8, pp. 63–92, 2013.
<https://doi.org/10.4203/ctr.8.3>
- [22] Csébfalvi, A., Csébfalvi, G. "Fair Comparison of Population-based Heuristic Approaches: The Evils of Competitive Testing", In: Proceedings of the 4th International Joint Conference on Computational Intelligence, Barcelona, Spain, 2012, pp. 306–309.
<https://doi.org/10.5220/0004168403060309>
- [23] Danka, S., Csébfalvi, A. "A Hybrid Metaheuristic for Project Scheduling Problems with Fuzzy Activity Durations to support the Ős-Dráva Water Management Programme", In: Proceedings of the Eighth International Conference on Engineering Computational Technology, Dubrovnik, Croatia, 2012, pp. 1–20.
<https://doi.org/10.4203/ccp.100.65>
- [24] Csébfalvi, A. "A new theoretical approach for robust truss optimization with uncertain load directions", *Mechanics Based Design of Structures and Machines*, 42(4), pp. 442–453, 2014.
<https://doi.org/10.1080/15397734.2014.880064>
- [25] Csébfalvi, A. "A New Algorithm for Robust Truss Optimization: Benchmark Results", In: Twelfth International Conference on Computational Structures Technology, Stirlingshire, UK, 2014, pp. 1–14.
<https://doi.org/10.4203/ccp.106.174>
- [26] Kaveh, A., Talatahari, S. "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", *Computers and Structures*, 87(5–6), pp. 267–283, 2009.
<https://doi.org/10.1016/j.compstruc.2009.01.003>
- [27] Kalatjari, V. R., Talebpour, M. H. "An Improved Ant Colony Algorithm for the Optimization of Skeletal Structures by the Proposed Sampling Search Space Method", *Periodica Polytechnica Civil Engineering*, 61(2), pp. 232–243, 2017.
<https://doi.org/10.3311/PPci.9153>
- [28] Sziveri, J., Topping, B. H. V., Iványi, P. "Parallel Transient Non-Linear Analysis of Reinforced Concrete Plate", In: Topping, B. H. V. (ed.). *Advances in Computational Structures Technology*, Civil-Comp Press, Edinburgh, UK, 1996, pp. 383–395.
<https://doi.org/10.4203/ccp.38.8.6>
- [29] Iványi, P., Topping, B. H. V. "Parallel and distributed dynamic relaxation form-finding", In: Topping, B. H. V. (ed.). *Innovation in Computer Methods for Civil and Structural Engineering*, Civil-Comp Press, Edinburgh, UK, 1997, pp. 157–165.
<https://doi.org/10.4203/ccp.50.10.2>
- [30] Sziveri, J., Topping, B. H. V., Iványi, P. "Parallel Dynamic Non-Linear Analysis of Reinforced Concrete Plates", In: Thierauf, G. (ed). *Engineering Structures Under Earthquake Loading*, Civil-Comp Press, Edinburgh, UK, 1997, pp. 69–84.
<https://doi.org/10.4203/ccp.51.5.1>
- [31] Obiała, R., Iványi, P., Topping, B. H. V. "Genetic algorithms applied to partitioning for parallel analyses using geometric entities", In: Bathe, H. J. (ed). *Computational Fluid and Solid Mechanics 2003*, Elsevier Science Publishers, Amsterdam, Netherlands, 2003, pp. 2078–2081.
<https://doi.org/10.1016/B978-008044046-0.50510-8>

- [32] Obiala, R., Seed, G. M., Topping, B. H. V., Iványi, P., Clar, D. E. R. "Genetic algorithm transformations for non-orthogonal models", In: Bathe, H. J. (ed). *Computational Fluid and Solid Mechanics 2005*, Elsevier, 2005, pp. 1275–1278.
- [33] Topping, B. H. V., Iványi, P. "Parallel, Distributed and Grid Computing for Engineering" (Computational Science, Engineering & Technology Series; Volume 21.), Saxe-Coburg Publications, Stirling, UK, Scotland, 2009.
- [34] Topping, B. H. V., Iványi, P. "Trends in Parallel, Distributed, Grid and Cloud Computing for Engineering" (Computational Science, Engineering & Technology Series), Saxe-Coburg Publications, Stirling, UK, Scotland, 2011.
- [35] Topping, B. H. V., Iványi, P. "Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering", Civil-Comp Press, Stirling, UK, Scotland, 2011. <https://doi.org/10.4203/ccp.95>
- [36] Topping, B. H. V., Iványi, P. (Eds). "Developments in Parallel, Distributed, Grid and Cloud Computing for Engineering (Computational Science, Engineering and Technology Series 31.)", Saxe-Coburg Publications, Stirling, UK., Scotland, p. 311, 2013.
- [37] Topping, B. H. V., Iványi, P. (Eds). "Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering", Civil-Comp Press, Stirling, UK, Scotland, 2013.
- [38] Iványi, P. "Parallel conversion of finite element meshes", *Pollack Periodica*, 9(3), pp. 89–102, 2014. <https://doi.org/10.1556/Pollack.9.2014.3.10>
- [39] Iványi, P. "A végeelem hálók párhuzamos konverziója", In: *Informatika a felsőoktatásban*, Debreceni Egyetem Informatikai Kar, Debrecen, 2014, pp. 30–38. (in Hungarian)
- [40] Iványi, P., Radó, J. "Előfeldolgozás párhuzamos számításokhoz", Typotex Kiadó, Budapest, Hungary, 2014.
- [41] Iványi, P. "Parallel Implementation of Dynamic Relaxation with CUDA", In: *Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, Pécs, Hungary, 2017, Paper 35.
- [42] Iványi, P., Kruijs, J., Kozubek, T., Gentzsch, W., Topping, B. H. V. "Civil-Comp: Parallel, Distributed and Cloud Computing", *Advances in Engineering Software*, 103, p. 12, 2017. <https://doi.org/10.1016/j.advengsoft.2016.11.002>
- [43] Iványi, P. "Parallel Implementation of Dynamic Relaxation with CUDA", In: *Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, Pécs, Hungary, 2017, p. 13. <https://doi.org/10.4203/ccp.111.35>
- [44] Iványi, P. "CUDA accelerated implementation of parallel dynamic relaxation", *Advances in engineering software*, 125, pp. 200–208, 2018. <https://doi.org/10.1016/j.advengsoft.2018.02.008>
- [45] Nvidia, "CUDA Zone", Nvidia, 2018. [Online]. Available: <https://developer.nvidia.com/cuda-zone>. [Accessed 15 04 2018].
- [46] Zhang, L., Quigley, S. F., Chan, A. H. "A fast scalable implementation of the two-dimensional triangular Discrete Element Method on a GPU platform", *Advances in Engineering Software*, 60–61, pp. 70–80, 2013. <https://doi.org/10.1016/j.advengsoft.2012.10.006>
- [47] Chau, M., Couturier, R., Bahi, J., Spiteri, P. "Asynchronous grid computation for American options derivatives", *Advances in Engineering Software*, 60–61, pp. 136–144, 2013. <http://dx.doi.org/10.1016/j.advengsoft.2012.06.005>
- [48] Chau, M., Garcia, T., Spiteri, P. "Asynchronous grid computing for the simulation of the 3D electrophoresis coupled problem", *Advances in Engineering Software*, 60–61, pp. 111–121, 2013. <https://doi.org/10.1016/j.advengsoft.2012.11.010>
- [49] Jariyasunant, J., Mai, E., Sengupta, R. "Algorithm for Finding Optimal Paths in a Public Transit Network with Real-Time Data", *Transportation Research Record: Journal of the Transportation Research Board*, 2256(34–42), pp. 34–42, 2011.
- [50] Tan, M.-C., Tong, C., Wong, S., Xu, J.-M. "An algorithm for finding reasonable paths in transit networks", *Journal of Advanced Transportation*, 41(3), pp. 285–305, 2007.
- [51] Bocz, P., Kisgyörgy, L., Vasvári, G. "Etraffic – an Open Access Transportation Model", *Periodica Polytechnica Civil Engineering*, 61(3), pp. 564–568, 2017. <https://doi.org/10.3311/PPci.9355>
- [52] D'Acerno, L., Montella, B., Lucia, F. D. "A Stochastic Traffic Assignment Algorithm Based on Ant Colony Optimisation", In: *Ant Colony Optimization and Swarm Intelligence*, Springer, Heidelberg, Germany, 2006, pp. 25–36. https://doi.org/10.1007/11839088_3
- [53] Hausman, J., McFadden, D. "Specification Tests for the Multinomial Logit Model", *Econometrica*, 52(5), pp. 1219–1240, 1984. <https://doi.org/10.2307/1910997>
- [54] Kaveh, A., Mahdavi, V. R., Kamalinejad, M. "Optimal Design of the Monopole Structures Using the CBO and ECBO Algorithms", *Periodica Polytechnica Civil Engineering*, 61(1), pp. 110–116, 2017. <https://doi.org/10.3311/PPci.8546>
- [55] Csébfalvi, A. "Robust Topology Optimization: A New Algorithm for Volume-constrained Expected Compliance Minimization with Probabilistic Loading Directions using Exact Analytical Objective and Gradient", *Periodica Polytechnica Civil Engineering*, 61(1), pp. 154–163, 2017. <https://doi.org/10.3311/PPci.10214>
- [56] Kaveh, A., Beitollahi, A., Mahdavi, V. R. "Locating Emergency Facilities Using the Weighted k-median Problem: A Graph-metaheuristic Approach", *Periodica Polytechnica Civil Engineering*, 62(1), pp. 200–205, 2018. <https://doi.org/10.3311/PPci.10665>
- [57] Katona, G., Juhász, J. "Autonomous cars in a multimodal transport plan", In: *Advanced Manufacturing and Repair Technologies in Vehicle Industry*, Zielona Góra, University of Zielona Góra, Faculty of Mechanical Engineering, 2018, pp. 137–149.
- [58] Cocchia, A. "Smart and Digital City: A Systematic Literature Review", In: *Dameri, R. P., Rosenthal-Sabroux, C. (Eds). Smart City*, Springer, Cham, 2014, pp. 13–43. https://doi.org/10.1007/978-3-319-06160-3_2
- [59] Hawkins, A. J. "Nvidia says its new supercomputer will enable the highest level of automated driving", 10 10 2017. [Online]. Available: <https://www.theverge.com/2017/10/10/16449416/nvidia-pegasus-self-driving-car-ai-robotaxi>. [Accessed 09 06 2018].

- [60] Schweber, B. "The Autonomous Car: A Diverse Array of Sensors Drives Navigation, Driving, and Performance", 2018. [Online]. Available: <https://hu.mouser.com/applications/autonomous-car-sensors-drive-performance/>. [Accessed 09 06 2018].
- [61] Stewart, J. "Self-Driving Cars Use Crazy Amounts of Power, and It's Becoming a Problem", 06. 02. 2018. [Online]. Available: <https://www.wired.com/story/self-driving-cars-power-consumption-nvidia-chip/>. [Accessed 09 06 2018].
- [62] Hruska, J. "Tesla Working With AMD on Self-Driving Car Processor", 21. 09. 2017. [Online]. Available: <https://www.extremetech.com/extreme/256171-tesla-working-amd-self-driving-car>. [Accessed 09 06 2018].
- [63] Teschler, L. "Centralized processing for autonomous vehicles", 07. 08. 2017. [Online]. Available: <https://www.microcontrollertips.com/centralized-processing-autonomous-vehicles/>. [Accessed 09 06 2018].
- [64] Dorigo, M. "Optimization, learning and natural algorithms", Ph.D. Thesis, Politecnico di Milano, Italy, 1992. (in Italian)
- [65] Dorigo, M., Blum, C. "Ant colony optimization theory: A survey", *Theoretical Computer Science* 344(1–2), pp. 243–278, 2005. <https://doi.org/10.1016/j.tcs.2005.05.020>
- [66] Kovács, G. "System model of electronic freight and warehouse exchange", PhD These, Budapest University of Technology and Economics, 2011.
- [67] Nassi, I., Shneiderman, B. "Flowchart techniques for structured programming", SIGPLAN Notices, New York, 1973.
- [68] Golomb, S. W., Baumert, L. D. "Backtrak programming", *Journal of the ACM*, 12(4), pp. 516–524, 1965.
- [69] Spiess, H., Florian, M. "Optimal strategies: a new assignment model for transit networks," *Transportation Research Part B: Methodological*, 23(2), pp. 83–102, 1988.