Parallel Methods of Solving of Supply Problem Using Ant Colony Optimization

Stanislav Piecka*
Department of Computer Science
Faculty of Management Science and Informatics
University of Žilina
Univerzitná 8215/1, 010 26 Žilina, Slovakia
piecka@gmail.com

Abstract

Presented work focuses on solving of the vehicle routing problem, which is considered as a basic supply problem. The main motivation is to achieve high quality solutions of the vehicle routing problem in the shortest possible time. It tries to achieve this goal via parallelization of the ant colony optimization metaheuristics, which is suitable for calculating of combinatorial problems. The vehicle routing problem is an NP-complete problem, therefore usage of heuristics is the only way to solve it on large instances.

The author proposes and compares different settings and solving methods of the vehicle routing problem using the ant colony optimization in sequence and parallel execution. He determines appropriate application of the local search methods, which improves solution quality and applies the elite approach. On selected communication topology, he proposes and compares several synchronization and communication strategies, all from achieved quality and speedup point of view, on different number of multi-core processors.

The author presents his own optimization tool, which is based on execution of the ant colony optimization in a parallel environment. The author applies proposed parallel method to the very large scale vehicle routing problem instances with as many as 1200 customers.

Categories and Subject Descriptors

D.1.3 [**Programming Techniques**]: Concurrent Programming; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

ant colony optimization, metaheuristics, parallel algorithm, multi-core architecture, combinatorial optimization, vehicle routing problem, very large scale vehicle routing problem

1. Introduction

Optimization of the transport system problems is currently required by science community and also by transport and telecommunication companies. There is a necessity of achieving high quality solutions of combinatorial problems in a very short time. The goal of research in this field is to find effective method to find solution of supply problems. Looking for optimal routes of larger problems with many customers is too expensive and therefore using of heuristic method is the only option. There exists several heuristic methods which can be used. One of them is called ant colony optimization (ACO). This modern method has been inspired by behaviour of ants and was discovered in late nineties. This method is used by scientists and there are many publications which discuss it. On its base there were developed many applications. This method was successfully used to solve NP-complete problems, where to get solution requires exponential time. These problems include: travelling salesman problem, quadratic assignment problem, solving of many transport problems, schedule problems, telecommunication problems, protein folding problems and so on. Modern computing technologies bring new opportunities and thesis for research. Currently available multi-core architecture allows running parallel computations also on personal computers. It makes sense to design new optimization methods and investigate existing parallel algorithms for this new computer architectures. In our work we have focused on possibilities of effective parallel implementation of the ant colony optimization method which is applied on vehicle routing problem. Ant colony optimization and its variations are suitable for using in parallel computing environment including multi-core architectures.

2. Vehicle Routing Problem Formulation

Symmetric vehicle routing problem can be according to [5] defined on complete not oriented graph $G = (V, E)$, where $V = \{0, ..., n\}$ is set of vertexes. Each vertex $i \in V \setminus \{0\}$ represents customer with positive demand q_i which is required by customer, where 0 means depot and $q_0 = 0$.

*Recommended by thesis supervisor: Assoc. Prof. Mária Lucká. Defended at Faculty of Management Science and Informatics, University of Žilina on September 16, 2012.

Piecka, S. Parallel Methods of Solving of Supply Problem using Ant Colony Optimization. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 5, No. 1 (2013) 1-12

Each edge $e \in E = \{(i,j) : i,j \in V, i < j\}$ has assignment positive value c_{ij} representing distance respectively costs required to transfer from i a j. There are m available vehicles with capacity Q in depot. It is required to determine m routes with minimal costs with following constrains:

- each route starts and ends in depot,
- each customer is served exactly once,
- sum of all satisfied demands in one route does not exceed vehicle capacity Q.

The solution can be imagined as set of arcs on graph which share exactly one point which is depot. To describe vehicle routing problem we introduce binary decision variable x_{ij}^k with following representation:

$$x_{ij}^k = \begin{cases} 1 & \text{, when vehicle } k \text{ follows vertex } j \\ & \text{immediately after vertex } i, \\ 0 & \text{otherwise.} \end{cases}$$

Problem is defined as

$$minf(x) = \sum_{i=0}^{n} \sum_{\substack{j=0 \\ j \neq i}}^{n} \sum_{k=1}^{m} c_{ij} x_{ij}^k \tag{1}$$

with following constrains:

$$\sum_{i=0}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij}^k q_j \leq Q \qquad 1 \leq k \leq m \tag{2}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ij}^k - \sum_{\substack{l=0 \\ l \neq j}}^{n} x_{jl}^k = 0 \qquad 1 \leq k \leq m, 0 \leq j \leq n \tag{3}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{k=1}^{m} x_{ij}^k = 1 \qquad 1 \leq j \leq n \tag{4}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij}^k \leq |S| - 1 \qquad \begin{matrix} \forall S \subseteq \{1, ..., n\}, \\ 1 \leq k \leq m, |S| \geq 2 \end{matrix} \tag{5}$$

$$x_{ij}^k \in \{0,1\} \qquad \begin{matrix} 1 \leq k \leq m, 0 \leq i \leq n, \\ 0 \leq j \leq n, i \neq j \end{matrix} \tag{6}$$

Constrains (2) ensures that vehicles can not be overloaded. Constrains (3) ensures that each customer visited by vehicle is also left by the same vehicle. Constrains (4) ensures that each customer is visited exactly once and depot is left by all used vehicles. Constrains (5) ensures that no sub-cycle is created in graph and obligatory constrains (6) ensures variables to be binary.

3. Ant Colony Optimization Method

Behaviour of single ant can be considered too simple to solve complex problems. But colony of ants can solve more complex problems which can not be handled by individuals. The ability of colony to find short path from food source to the nest is an example. Single ant can not determine itself whether selecting of route is profitable. This phenomenon was in [15] explained by ability of ants to communicate indirectly using pheromone on the ground. This approach is basic for the ant colony optimization method. It was shown in [8] how approach of combination of heuristic information and probabilistic behaviour of ants can be used to optimize combinatorial problems. Computation of problem which was looking for shortest path was described by following artificial ant model:

- the artificial ant creates solution by travelling between vertexes and in each vertex ant decides to choose next vertex to be visited by probability specified by amount of the pheromone,
- the ant remembers visited vertexes to avoid subcycles,
- when food is found by ant this ant returns by deterministic way to the nest and leaves pheromone trail on travelled edges,
- the intensity of left pheromone is given by achieved quality of constructed solution,
- the pheromone on each edge evaporates in time.

Optimization of ant colony can be according to [12] applied to each combinatorial problem which can be defined by constructive heuristics.

First published algorithm which used ant system (AS) for solving of the travelling salesman problem was presented in [11]. During initialization phase of algorithm is sets matrix of pheromone to value τ_0 according to following formula: $\tau_{ij} = \tau_0 = \frac{m}{C^{nn}}, \forall (i,j)$, where C^{nn} is length of route produced by nearest neighbourhood heuristics.

During creating of solutions ant k which is in vertex i decides to visit vertex j with following probabilistic rule:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} & \text{when } j \in N_i^k, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\eta_{ij} = \frac{1}{d_{ij}}$, while d_{ij} is distance between vertexes i and j, α and β are parameters which specify influence of pheromone and heuristic value and define balance between exploitation and exploration. N_i^k denotes set of feasible neighbourhood of vertex i where the ant k is.

After finishing of each iteration pheromone evaporates from each edge independently on behaviour of ants. This can be described by following equation:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \ \forall (i,j) \in L \tag{8}$$

where ρ means pheromone evaporation coefficient.

When construction of feasible solution is finished all m ants alter pheromone matrix by following equation:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k, \ \forall (i,j) \in L, \tag{9}$$

where $\Delta\tau_{ij}^k$ is amount of pheromone put by ant k on edge (i,j) and is defined by:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C^k} & \text{when edge } (i,j) \in T^k, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where C^k is length of route T^k created by ant k and it is calculated as summary of all lengths of all edges which belongs to T^k. Equation (10) ensures that better solutions alter pheromone matrix with higher influence. This way edges which are in good solutions are more preferable in next iterations.

It was shown in [12] that the ant system method converges to optimum. But the biggest disadvantage is dramatically reduced performance according to increased number of vertexes.

3.1 Rank-based Ant System (AS$_{rank}$)

Improvement of the ant system which was presented in [3] is combination of principles of elite strategy and graded putting of pheromone on the ground. At the end of each iteration prior to pheromone evaporation all ants are ordered by achieved quality. Each elitists ant alters matrix of pheromone according to its own order in elite r. The formula of pheromone actualization (9) is modified to:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w-1} (w-r)\Delta\tau_{ij}^{r} + w\Delta\tau_{ij}^{bs}, \ \forall (i,j) \in L, \quad (11)$$

where w is size of elite, $\Delta\tau_{ij}^{r} = \frac{1}{C^r}$ and $\Delta\tau_{ij}^{bs} = \frac{1}{C^{bs}}$, while C^r is length of route produced by elitist ant r and C^{bs} is length of best so far solution. Both equations (8) and (7) are unchanged.

Results presented in [3] shows that AS$_{rank}$ produces better results that simple elitists ant system (EAS) presented in [11] and much better results as the ant system. This is achieved especially when larger instances are solved.

3.2 Savings Based Method

There was adapted algorithm AS$_{rank}$ to solve of vehicle routing problem in [4]. In [20] this algorithm was improved by applying of local search method which was applied after construction of solutions of ants. Algorithm [20] leaves equations of pheromone modification (8) and (11). It leaves also probabilistic rule (7) but estimation of costs η_{ij} is according to [6] defined as $\eta_{ij} = d_{0i} + d_{j0} - d_{ij}$, where d_{ij} represents distance between vertexes i and j, and index 0 represents depot (see Figure 1). On contrary of [4] which applies 2-Opt heuristics over each produced solution, method presented in [20] prior to 2-Opt heuristics tries to swap customers between different routes.

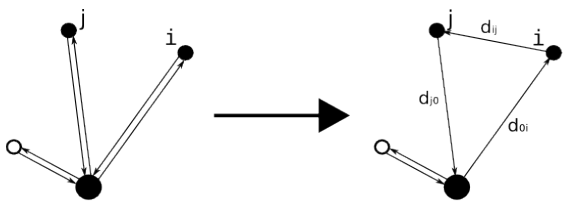

Figure 1: Visualization of selecting the savings η_{ij}

In this work we did experiments comparing results from different constructing methods which are shown in Table 1. We have used the best one denoted CVRPs+LS in next experiments.

3.3 Influence of the Elite Size

There is defined parameter w denoting the elite size in equation (11). In separate experiment we have tested influence of the elite size on our implementation of saving based ant colony optimization. We have changed the elite size from value 1 to 12 and also for value 99. We have found appropriate value of the elite size to be 6 ants which we have used in following experiments. The value is similar to one presented in [10]. In Figure 2 we can see differences in convergence speed according to the elite size for testing instances [5] and [14].

4. Application of local search

Application of the local search method markedly improves results of the ant colony optimization. Therefore we have decided to compare several suggested local search strategies to achieve best possible ratio of solution quality and shorting of execution time. There follows tested strategies in parallel environment as they are also presented in Table 2:

a) Without application – in this case there are no changing in achieved solutions and the elite is composed from the best solutions at the end of each iteration.

b) With swap of customers between different routes over each solutions – exchanging of customers between different routes is done over each solution produced by ants. The elite is chosen at the end of each iteration from improved solutions.

c) With swap of customers between different routes and 2+2.5-Opt over solutions which are better than 90% of previously obtained solutions – the first iteration does both improvements over each solution. At the end of iteration it is determined how much better is the value of best solution of iteration. Percentage value of the improved solution is multiplied by coefficient 0.9 and it is remembered. In the following iteration the improvements are applied only on solutions which achieved quality is better than value of already known best solution decreased by remembered improvement.

d) With swap of customers between the routes over each solution and 2+2.5-Opt over elite solutions at the end of the iteration – it tries to exchange customers between routes over each solution. The 2 and 2.5-Opt heuristics is applied only over achieved elite.

e) With swap of customers between different routes and 2+2.5-Opt over all solutions – all improving procedures are applied over each solution. There is no other operation done over the elite which is given by best achieved solutions.

f) With swap of customers between different routes and 2+2.5-Opt over elite solutions at the end of the iteration – in this case there are all improving procedures applied at the end of iteration over the achieved elite.

g) With swap of customers between different routes and 2+2.5-Opt over the 10% of the best solutions at the end of creation of solutions – there are applied local search procedures over 10% of best solutions at the end of each iteration. There is improved chance of good solution to get into elite. The elite is determined after this step.

	C4	C5	G18	G19	G20
CVRP1	1669.91±8.80	2075.02±14.69	1290.28±6.21	1811.74±9.11	2442.65±12.14
CVRP1+LS	1426.34±8.48	1813.41±12.77	1177.30±4.00	1628.59±10.88	2209.43±4.58
CVRP2	1700.70±17.39	2191.40±26.83	1549.35±22.76	2178.33±31.12	3135.35±18.37
CVRP2+LS	1181.44±14.39	1482.57±15.06	1071.80±2.73	1472.84±6.88	1937.41±4.13
CVRPs	1073.64±12.32	1375.77±10.20	1090.18±7.17	1490.94±10.26	2005.39±14.27
CVRPs+LS	1059.67±4.80	1334.43±10.66	1048.37±3.72	1434.79±5.04	1920.52±12.99

Table 1: Comparison of solution quality of selected tested instances by the multiple travelling salesman problem with coefficient $\eta_{ij} = \frac{1}{d_{ij}}$ (denoted as CVRP1), $\eta_{ij} = d_{i0} + d_{0j} - 2 * d_{ij} + 2 * |d_{i0} - d_{0j}|$ (denoted as CVRP2) and saving based method (denoted as CVRPs); with and without using of local search 2+2.5-Opt

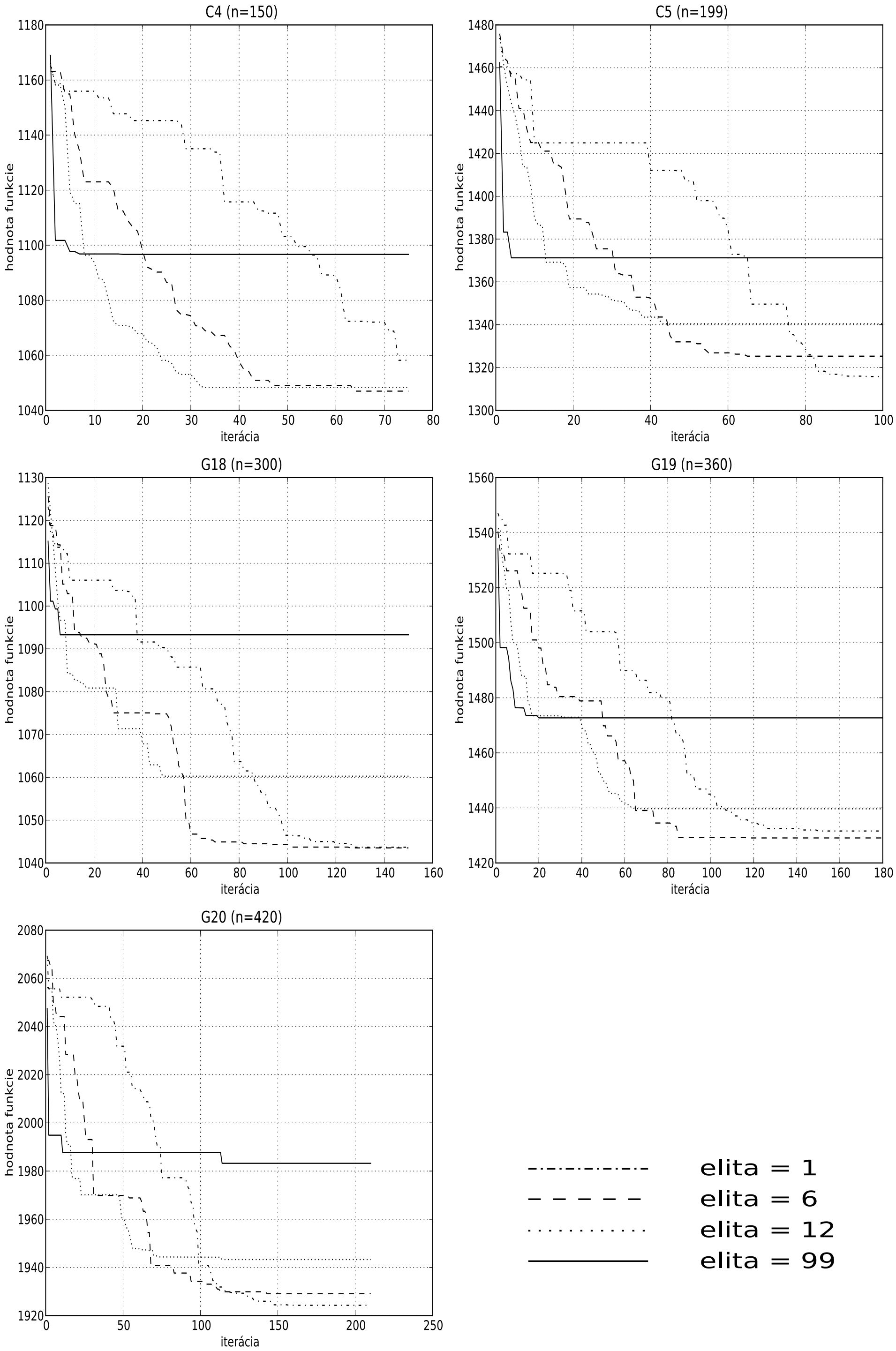

Figure 2: Speed of convergence according to size of elite for presented instances.

instance and	strategy	1 core		2 cores		4 cores		8 cores / 2 nodes		16 cores / 4 nodes		32 cores / 8 nodes	
		Q	$t_r[s]$	Q	$t_r[s]$	Q	$t_r[s]$	Q	$t_r[s]$	Q	$t_r[s]$	Q	$t_r[s]$
	a)	1073.64	66.88	1076.77	67.86	1072.82	68.91	1065.91	70.26	1064.48	69.63	1061.25	70.37
	b)	1072.66	90.18	1065.60	90.68	1062.45	91.56	1060.96	94.15	1057.52	93.29	1053.89	91.99
	c)	1068.51	69.13	1071.80	69.45	1066.96	70.45	1066.38	70.95	1063.51	71.36	1058.08	71.29
C4	d)	1061.60	88.83	1055.27	89.22	1053.67	91.28	1052.58	90.78	1053.69	91.78	1051.36	91.93
	e)	1058.17	92.02	1055.33	92.35	1055.20	93.97	1052.56	93.91	1051.50	94.40	1049.12	93.99
	f)	1064.90	67.90	1060.88	68.35	1059.23	69.59	1060.19	70.23	1053.44	70.13	1051.62	70.02
	g)	1059.67	68.80	1059.23	69.60	1054.04	70.85	1052.17	70.94	1048.28	71.41	1049.66	71.14
	a)	1375.77	237.93	1371.64	241.20	1364.04	240.64	1353.21	244.32	1349.21	247.45	1342.27	247.35
	b)	1357.70	326.04	1349.92	330.03	1351.60	330.62	1340.79	332.55	1339.25	332.17	1336.04	331.51
	c)	1356.29	248.86	1353.31	247.83	1350.67	252.76	1351.95	255.05	1345.06	254.16	1347.97	258.79
C5	d)	1335.41	320.09	1335.96	323.22	1335.29	324.57	1331.41	330.10	1326.90	330.67	1323.90	328.21
	e)	1343.68	325.78	1337.58	330.46	1333.51	329.91	1327.99	334.72	1325.24	335.16	1321.48	336.61
	f)	1342.95	240.51	1340.17	243.60	1336.99	247.12	1334.55	249.76	1335.50	248.54	1331.07	249.49
	g)	1334.43	245.89	1336.36	249.34	1328.61	251.02	1326.73	255.08	1319.57	254.37	1318.83	256.76
	a)	1090.18	1531.56	1083.51	1562.11	1077.33	1566.82	1079.53	1541.82	1070.51	1590.65	1067.67	1583.74
	b)	1068.37	2330.68	1062.82	2380.51	1062.56	2418.77	1058.93	2409.50	1052.99	2431.21	1045.54	2434.68
	c)	1070.89	1680.53	1060.34	1687.63	1064.75	1714.85	1054.70	1725.84	1056.34	1716.97	1053.24	1749.00
G18	d)	1044.24	2270.03	1050.93	2273.50	1044.15	2340.83	1044.12	2342.69	1038.11	2381.55	1032.84	2387.60
	e)	1044.53	2309.11	1048.31	2363.01	1046.34	2366.81	1040.62	2387.26	1037.65	2368.95	1033.82	2418.15
	f)	1054.07	1517.16	1050.27	1553.18	1045.99	1600.20	1046.14	1588.03	1044.23	1605.39	1038.50	1626.01
	g)	1048.37	1601.62	1043.67	1611.73	1045.53	1615.54	1039.97	1674.77	1037.42	1671.72	1034.88	1671.67
	a)	1490.94	3549.70	1483.58	3594.26	1478.67	3673.24	1474.32	3688.81	1463.99	3809.60	1461.26	3783.59
	b)	1458.85	5733.60	1452.60	5715.61	1450.66	5808.61	1447.15	5858.10	1438.95	5964.54	1436.29	6014.94
	c)	1469.92	4053.79	1458.32	4079.17	1455.56	4122.13	1445.12	4234.11	1453.05	4261.10	1441.35	4306.77
G19	d)	1434.23	5486.13	1431.62	5567.16	1431.34	5682.22	1426.92	5685.51	1422.85	5784.89	1422.95	5816.71
	e)	1435.61	5591.57	1431.83	5689.66	1427.49	5854.06	1423.01	5768.84	1424.26	5806.44	1418.52	5886.48
	f)	1435.20	3638.56	1427.62	3686.58	1432.86	3782.66	1422.08	3931.78	1418.91	3932.88	1424.02	3946.10
	g)	1434.79	3729.62	1434.52	3700.21	1430.27	3854.76	1423.27	3834.98	1419.78	3983.18	1419.06	4001.67
	a)	2005.39	7319.65	1984.39	7288.09	1981.95	7414.20	1967.42	7482.85	1965.96	7492.99	1952.96	7659.46
	b)	1946.34	12222.55	1942.88	12529.92	1934.30	12965.74	1930.61	12846.25	1921.07	13241.31	1918.47	13040.75
	c)	1948.22	8414.95	1945.18	8517.72	1947.83	8701.51	1947.39	8742.08	1940.89	8717.29	1938.78	8866.79
G20	d)	1915.45	12038.82	1912.30	12215.34	1909.23	12431.00	1905.42	12610.59	1898.83	12790.59	1895.68	12953.67
	e)	1906.72	12128.04	1908.05	12435.38	1913.82	12387.86	1901.70	12857.91	1899.71	12834.34	1894.33	13048.26
	f)	1916.87	7615.33	1916.62	7652.06	1910.59	7659.71	1905.30	8007.52	1903.83	7982.19	1893.48	8184.40
	g)	1920.52	7645.47	1916.22	7832.38	1909.70	7971.88	1901.06	8175.61	1896.16	8373.35	1891.70	8355.33

Table 2: Comparison of different local search methods and strategies of their application in parallel environment for specified instances, where t_r means running time and Q means average value of obtained quality.

From the results shown in Table 2 we can see that the best quality is provided by running of all methods of local search over each solution. But this strategy is very time consuming. Fastest results were provided when no improving procedures was applied but this method gave the worst results. We can identify long-time strategies: b), d) and e) and short-time strategies: a), c), f) and g). The best ratio between quality of the solution and the calculation speed gives the strategy which applies local search at the end of iteration over the 10% of the best solutions. With 32 colonies it achieved similar quality of solutions as the best achieved solutions with instances C4, G18 and G19. The difference is smaller than 0.2%. This strategy achieved best results for instances C5 and G20. On the other hand the execution time was different from strategy without application of improving method by 1% in case C4, 3% in C5, 5% in G18, 6% in G19 and 9% in G20. Comparing the strategy e) which had almost the same quality for instance G20 it requires more than 70% of time.

5. Parallel Ant Colony Optimization

The ant colony optimization metaheuristics is characterized by possibility to execute relatively independent ants in parallel. The communication between ants is necessary only in specific situations. The term communication means exchanging of information needed by artificial ants for their execution. Typically the pheromone matrix or the best solution is exchanged. In case of AS_{rank} also elite solutions are exchanged. Received solutions must be arranged. Parallel running of the ant colony optimization is shown by Algorithm 3.

Taxonomy of the parallel ACO algorithm strategies was presented in [19] and it is shown in Figure 4.

```
DO PARALLEL
  Set up parameters
  Initialize of pheromone matrix
  WHILE (terminate condition not met)
    Create solutions by ants
    Apply local search // optional
    IF (communication step)
      Exchange information between colonies
      Evaluate received information
    ENDIF
    Update pheromone matrix
  ENDWHILE
END PARALLEL
```

Figure 3: Parallel execution of Ant Colony Optimization.

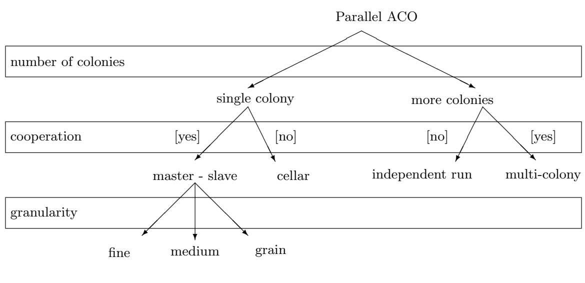

Figure 4: Taxonomy of parallel ACO algorithms.

Cluster which was used in our tests contained 32 cores divided in 8 computation nodes. Used communication topology was based on two levels. Within each node it was possible to use shared memory therefore all communication inside nodes were implemented in this way. There was also used low level caching between cores. Communication which was done between different nodes was based on User Datagram Protocol. Inter-node communication was based on "each to each" approach.

6. Proposed Communication Strategies

In this work we were looking for the best communication strategy, to produce high quality solutions in the shortest time. There were done two experiments on each communication strategy with 1, 2, 4, 8, 16 and 32 used cores on each tested problem instance.

At the first the number of the ants was fixed. Ants were divided between colonies, counting $2 * n$ iterations. The results compares how the algorithm behaves in the parallel environment when the number of the ants is fixed. The quality, the running time, communication/ waiting time and the speedup was evaluated for every strategy.

In the second experiment the number of the ants was set as the half of the number of customers in each colony. In this experiment, the termination rule was specified according to the time of execution, from the previous experiment. The exception was made with 32 colonies when we tried two different termination times. The aim of the second experiment was how the number of the ants with fixed finishing time influences the quality of the solution. In every experiment we have used settings from the analysis of sequence algorithm. The pheromone evaporate ratio was 0.95, the elite was 6 ants, the local search procedure was done on the best 10% of ants, $\alpha = \beta = 5$. The size of neighbourhood in saving list was $\frac{n}{2}$.

6.1 Algorithm without exchanging of information

The first researched parallel method in this work is the algorithm using the strategy when each colony does not exchange the information. Only final solutions of colonies are compared at the end of execution. It is possible to use this strategy only because the ant colony method uses randomness. It depends on fact that every colony decides to examines different part of search space in the first iterations. The algorithm as can be seen in Graph 5 does not give bad results even to its simplicity and the communication time is marginal.

6.2 Synchronous algorithm

The second experiment presents method which synchronously exchanges the best solution and elite solutions between ant colonies. It tries to keep actual information about the whole system between colonies. The colonies are waiting and exchanging the information about their elite's solutions at the end of every iteration. Each colony keeps its own copy of pheromone matrix so ants does not need to wait each time when they try to access it. The pheromone matrices of each colony are modified by exchanged information at the end of the iteration. The big disadvantage of this approach is high time required by synchronization process.

6.3 Asynchronous exchanging of the best solution

This algorithm tries to exchange the best solution only in case it was found instead of every iteration. Saved communication time is used to search for another solutions. Each colony manages own, differently influenced pheromone matrix. To reduce probability of convergence to the local optimum as discovered in [12] when only best solution is exchanged, we exchange several elite solutions of the ant colony. It is possible to accept whole elite in receiving colony or to combine received elite with the local one. We have discovered in experiment that slightly better results are given by the combination of received elite and local elite.

6.4 Asynchronous exchanging of the pheromone matrix

Following two algorithms use the pheromone matrix as the exchanging information between colonies. It is possible to exchange the pheromone matrix, combination of solutions and the pheromone matrix or the parts of the pheromone matrix. There was mentioned that from solution quality point of view there is no difference between synchronously sending pheromone matrix or best solutions in [12]. But instead of that we have used asynchronous communication. Each colony has its own pheromone matrix which can be different.

6.4.1 Influence by higher value of the pheromone

Following communication method has two characteristics. It exchanges complete pheromone matrix between the colonies and tries to avoid staying in the local optimum. Usually the pheromone matrix in AS_{rank} is evaporated according to (8). Therefore later the probability of creating solutions which have low level of pheromone on their edges is decreasing. Presented strategy influences the pheromone matrix as the whole unit to increase the chance of selecting several perspective edges from other colonies by additional increasing of the level of pheromone. Elements of local pheromone matrix in case of receiving another pheromone matrix are influenced by following equation:

$$\tau_{ij} \leftarrow max\{\tau_{ij}, \tau^r_{ij}\}, \tag{12}$$

where τ_{ij} represents pheromone value on arc (i, j) and τ^r_{ij} represents pheromone value on the same arc on received pheromone matrix.

The biggest disadvantage of the algorithm is the necessity of exchanging of complete matrix which size has higher order. Sharing memory between the colonies in one computing node and exchanging matrix in compressed form has been used to achieve higher performance. This time is counted to the time which is spent with communication.

6.4.2 The probability of influencing the pheromone

This method tries to improve previous one by two changes. It tries to decrease amount of exchanged data and tries to exchange only interesting elements of the pheromone matrix, because it is useless to exchange data which does not improve the quality of solutions of receiving colony. This is done by counting the average value of sending pheromone matrix and according to this probability, elements are or are not sent. The probability of sending the element of the pheromone matrix τ_{ij} is given:

$$p = \frac{\tau_{ij}}{\alpha \tau_{avg}}, \tag{13}$$

where τ_{avg} is the average value of the pheromone matrix and α is transfer coefficient which we have set empirically to value 2. The influence of elements in the pheromone matrix of receiving colony is done by equation (12).

The amount of exchanged data was decreased by this approach. The analysis of the variances did not show significant difference between obtained results in case we have executed fixed number of iterations. It could seem to be good improvement because of amount of exchanged data has been decreased but actually it is not. Calculating of average value and stochastic process of choosing elements to be sent rapidly increases execution time. This wasted time causes decreasing of solution quality in the second experiment terminated by execution time. This was also confirmed by the analysis of invariance.

6.5 Comparison of Achieved Results

From achieved results presented in Graph 5 and 6, we can see there is no outstanding winner. To compare the methods we have decided to do the average of all solutions of problem instances C4, C5, G18, G19 and G20. This comparison is shown in Table 3 for the experiment terminated after $2n$ iterations and in Table 4 for the experiment finished by execution time.

In the first case, best results of instances G18, G19 and G20 was gained by the strategy which exchanged elements of the pheromone matrix according to probability. Best solution of instance C4 was obtained by strategy exchanging the whole pheromone matrix and C5 by asynchronous method which exchanges the best solution and elite. The results of all three mentioned strategies was similar for instances C4 and C5. In case we have relatively small number of ants in each colony it is better to exchange information based on pheromone matrix. This was confirmed by the analysis of invariance which showed significantly better solution quality for the instances G18 ($F = 11.50$, $p \leq .01$), G19 ($F = 9.97$, $p \leq .01$) and G20 ($F = 23.71$, $p \leq .01$).

This was different in case of higher number of ants when it was better to exchange smaller amount of data. In second experiment where $\frac{n}{2}$ ants were assigned to each colony, which was terminated by execution time, methods which exchanged pheromone matrix were worse. For comparison of the influence of exchanging the pheromone matrix on solution quality we made analysis of invariance which showed that the quality of solution is significantly worse on smaller instances C4 ($F = 44.31$, $p \leq .01$), C5 ($F = 24.54$, $p \leq .01$) and G18 ($F = 5.41$, $p \leq .01$). This is caused by longer communication times which includes communication time, synchronization time, time for compressing pheromone matrix and the time for choosing the elements to be send from the pheromone matrix.

When higher number of the ants is used the best strategy is asynchronous exchange of the best solution and the elite between the colonies. This strategy produced the best average result and the best results for instances C4, G18, G19 and G20. Instance C5 had the best result by the synchronous algorithm exchanging the best solution and elite. Considering achieved results we can advise asynchronous method exchanging the best solution and elite also for the very large scale vehicle routing problems.

6.6 Evaluation of the Number of Ants Allocated to the Colony

From achieved results of communication strategies, we can see that better solutions are obtained when termination criterion based on execution time is used. This is caused by using $\frac{n}{2}$ artificial ants in each colony. Here more ants create more solutions and therefore chance to find good solutions is higher. Only exception is the situation with 32 cores. There we can see significantly worse results in each communication strategy as we expected. The higher number of ants requires longer time to compute each iteration. It could mean that the achieved results are significantly worse because the number of executed iterations in specified time is too small. To check this behaviour we made another test where the execution time was extended by 20%. In this case the quality of the solutions was better.

We analyzed results of experiment terminated by fixed number of iterations where the artificial ants were $\frac{n}{32}$ for each colony. In this case the quality of solution was not increasing during last iterations. The quality was increasing in case of the second experiment. Therefore it is important to find such number of the ants which is not too small but still big enough to stabilize pheromone matrix. This way we could obtain better results.

We did another experiment to search of the appropriate number of ants per colony in parallel environment where the number of ants was changed according to the number of customers and it is shown in Table 5 with achieved quality.

In the Table 5 we can see that the best obtained results are with $\frac{n}{4}$ artificial ants per each colony. With changing of the number of ants the solution is getting worse. Not enough iterations were done in cases of $\frac{n}{2}$ and $\frac{n}{3}$. But in case of $\frac{n}{8}$, $\frac{n}{16}$ and $\frac{n}{32}$ not enough solutions were examined by the small number or ants. The analysis of invariance showed that there is significant difference between the results obtained from different number of ants per colony for all tested instances C4 ($F = 21.88$, $p \leq .001$), C5 ($F = 47.54$, $p \leq .001$), G18 ($F = 42.95$, $p \leq .001$), G19 ($F = 56.35$, $p \leq .001$) and G20 ($F = 56.33$, $p \leq .001$). We have to consider the influence of the number of ants especially when short execution time is used.

7. The Very Large Scale Vehicle Routing Problems

When original method was used on the very large scale vehicle routing problem we did not achieve appropriate solution quality in required time. Therefore we applied sweep heuristics before execution of the ant colony optimization method. By affecting pheromone matrix by solution of sweep heuristics we achieved shorter computation time with increasing of solution quality.

The development of solution quality using sweep heuristics and without its usage for instances G21 to G32 is shown in Figure 7. It can be seen that there is obtained much better solution quality from the beginning of execution in case of applying the sweep heuristics.

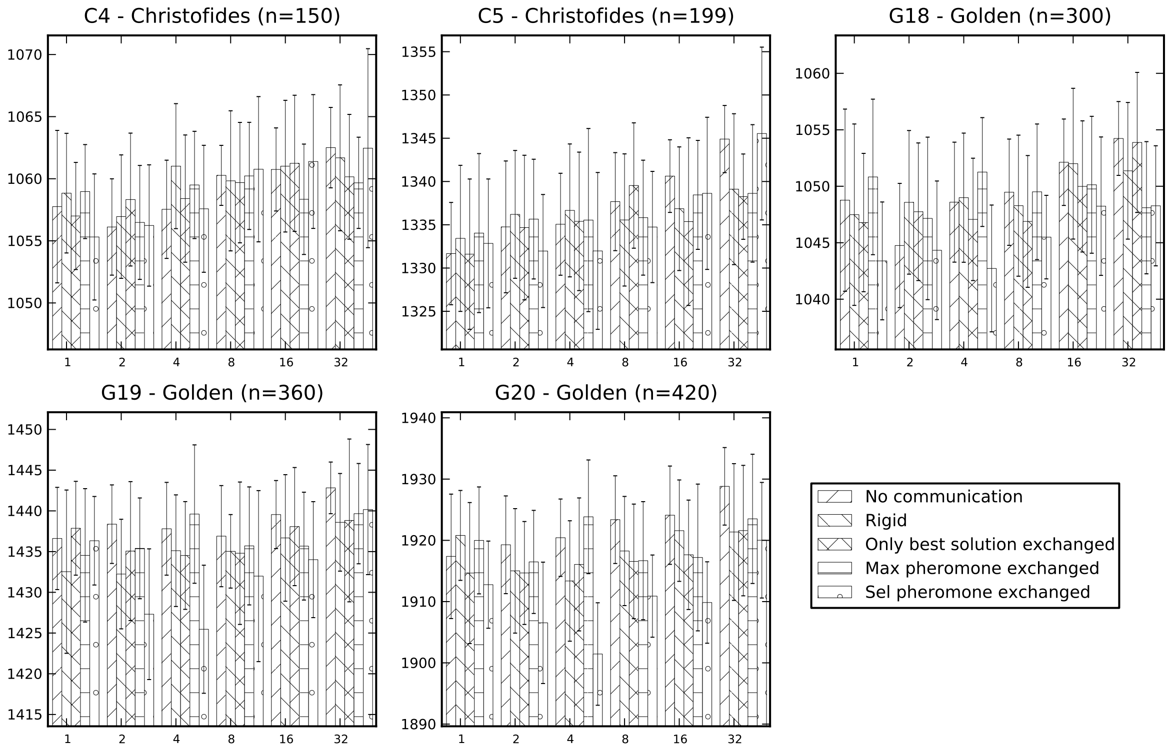

Figure 5: Comparison of solution quality of different communication strategies with different number of colonies when fixed number of iterations was executed. Number of ants is fixed system-wide.

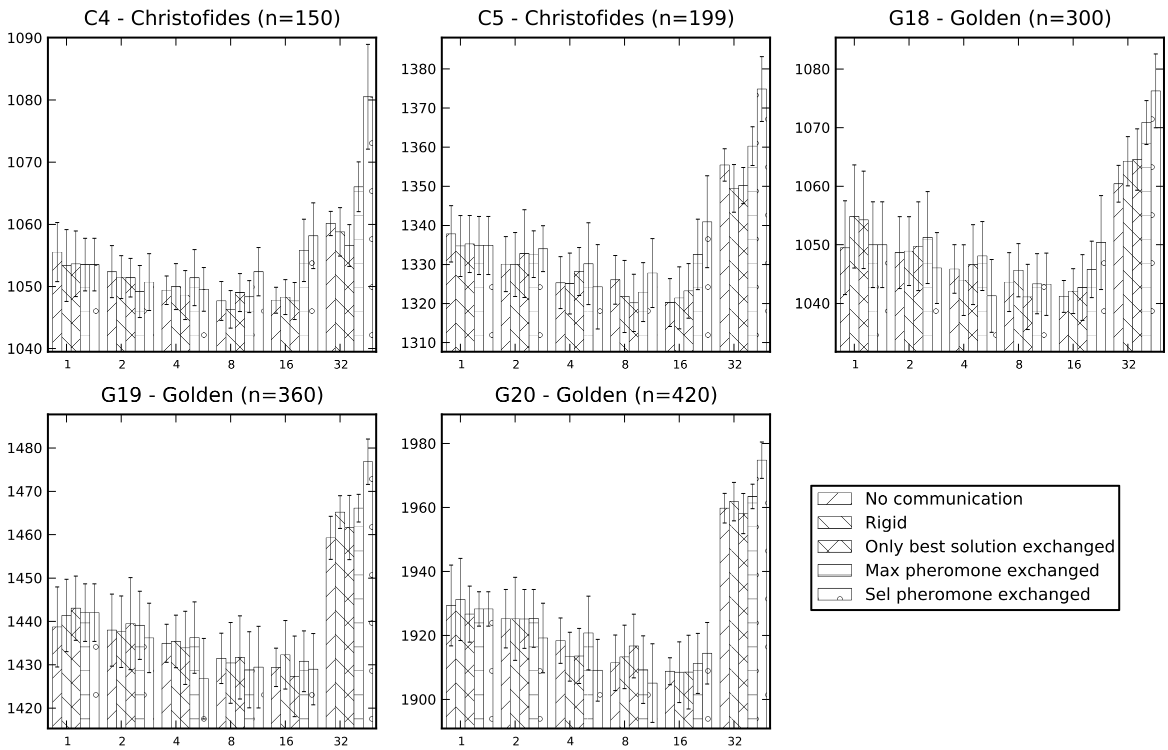

Figure 6: Comparison of solution quality of different communication strategies with different number of colonies when execution time is fixed. Each colony contains fixed number of ants.

instance	Q_{nc}	t_r	Q_r	t_r	Q_b	t_r	Q_{mp}	t_r	Q_{sp}	t_r
C4	1059.16±4.58	46.40	1059.89±5.55	46.25	1059.14±5.25	45.83	1058.87±4.39	46.04	1058.96±6.44	46.14
C5	1337.45±7.15	161.74	1336.30±8.07	162.32	1335.80±8.39	161.36	1336.36±8.27	160.71	1335.95±9.53	161.63
G18	1049.66±6.14	1043.65	1049.45±6.79	1047.59	1048.73±6.47	1047.94	1049.50±6.38	1043.37	1045.41±5.86	1042.31
G19	1438.68±5.61	2488.99	1435.03±7.55	2615.18	1436.53±8.17	2466.16	1436.76±7.53	2467.35	1432.55±9.46	2442.07
G20	1922.23±8.66	5258.03	1918.40±9.94	5145.81	1916.87±10.33	4993.67	1919.62±10.36	4986.45	1910.25±9.96	5034.63
average	1361.43±319.59	1799.76	1359.81±318.01	1803.43	1359.41±317.84	1742.99	1360.22±318.70	1740.78	1356.62±316.02	1745.36

Table 3: Comparison of average values of presented strategies and running times when n ants were divided across colonies, where Q denotes average quality of solution and t_r denotes running time based on executed number of iterations.

instance	Q_{nc}	t_r	Q_r	t_r	Q_b	t_r	Q_{mp}	t_r	Q_{sp}	t_r
C4	1052.16±5.58	46.21	1051.37±5.61	46.15	1051.16±4.88	46.20	1054.05±7.26	46.21	1057.47±11.93	46.20
C5	1332.50±13.23	162.31	1330.44±12.55	162.21	1331.65±12.37	162.31	1335.57±14.08	162.32	1339.46±19.06	162.35
G18	1050.22±8.07	1045.29	1049.96±9.58	1044.73	1049.83±10.34	1045.23	1051.56±10.86	1045.34	1051.22±13.46	1045.28
G19	1441.64±11.80	2535.24	1440.39±13.91	2533.94	1439.52±14.29	2535.13	1440.53±14.32	2535.21	1440.05±19.06	2535.19
G20	1925.52±18.90	5201.55	1925.59±20.72	5199.95	1924.81±18.67	5201.54	1926.42±20.04	5201.60	1925.18±25.28	5201.45
average	1360.41±322.60	1798.12	1359.55±322.60	1797.40	1359.39±322.31	1798.08	1361.63±322.02	1798.13	1362.68±321.13	1798.09

Table 4: Comparison of average values denoted by Q obtained by presented communication strategies and execution times t_r when $\frac{n}{2}$ artificial ants were assigned to each colony. Termination criterion is based on execution time.

In contrary of the D-Ant in [9], the sweep heuristics was not used for dividing routes with following calculating of the travelling salesman problems over each of route. It was used to achieve good starting solution and to focus ant colony to areas similar to sweep colony.

To determine appropriate amount of influence of the pheromone matrix another experiment was done. If we influenced the pheromone matrix too little we did not get any advantage. If we influenced the pheromone matrix too much the colony could not improve the solution and it quickly stopped in one of the local optimum. The results of different affection of starting pheromone matrix are presented in Table 6. The application is done by equation:

$$\tau_{ij} = \begin{cases} \tau_0 & \text{, when edge } (i,j) \text{ belongs to solution,} \\ \tau_0 * \eta & \text{otherwise.} \end{cases} \tag{14}$$

From the results of three different instances we can see the best results were done in the case of influencing by the $\eta = 0.7$ for tasks G21 and G32 with $\eta = 0.5$ for task G27. We expected that different values caused lower quality of the solution.

7.1 Obtained Solutions of the Very Large Scale Vehicle Routing Problem Instances

For solving of the very large scale vehicle routing problem in parallel environment we have used 32 cores divided into 8 nodes. Each core counted one colony where we empirically allocated $\frac{n}{16}$ artificial ants where n means the number of customers. The colonies were asynchronously cooperating and exchanging information about the best solution and the elite of the colony. Affection of the pheromone matrix was done by coefficient $\eta = 0.7$. We have used time based termination criterion which was given from previous experiment where n iterations were done. It was determined by time when the value of the solution did not improve any more.

Obtained solutions from the very large scale vehicle routing problem of ten independent runs are presented in Table 7. Almost in all instances we achieved average solution to be under 1% of the best-known solution which was estimated in [18]. Significantly bad results were achieved in instances G25 with 760 customers and G27 with 840 customers. This is caused because these two problems have another structure (see [18]) and the sweep heuristics does not produce appropriate division of customers into routes. The colonies affected by this wrong solution were not able to improve enough quality of solution considering to relatively short termination time.

8. Conclusions

We have developed the method using the ant colony optimization metaheuritic for solving vehicle routing problem in this work. We have chosen parallel environment to achieve quality solutions in the shortest possible time. We have analyzed two possible construction methods: greedy constructive heuristics and saving based method. We have compared obtained solutions. We have set up appropriate parameters of the method in parallel environment by using of literature and by our experiments. We have shown different local searching strategies and their influence on solution quality. We have presented our own optimization tool which was designed to solve vehicle routing problems in parallel environment by ACO. This tool is portable, configurable and free. It can be used in massive parallel systems and in multi-core personal computers as well. It can be easily improved for different problems. We have applied our parallel ant colony optimization method to several problem instances, with 150 - 420 customers. We have applied optimized algorithm on the vehicle routing problem with 560 - 1200 customers. We have shortened computation time by affecting pheromone matrix by solution obtained by sweep heuristics. We have determined appropriate ratio of pheromone matrix affection. The method was applied in parallel environment using double-level communication topology. We have analyzed efficiency and speedup of proposed method in parallel execution.

As far as we know, this is the first application of ant colony optimization method to instances of the very large scale vehicle routing problem. In [1] are presented high quality solutions obtained by genetic algorithm but with worse computation times. Solution of instance G21 (560 customers) lasted 10 hours in their experiment. We have obtained similar quality in 9.3 minutes in our experiments. Solution of instance G32 (1200 customers) lasts in [1] 74

instance	n/2	n/3	n/4	n/8	n/16	n/32
C4	1056.60±3.44	1048.19±3.77	1046.93±4.06	1048.01±3.96	1052.61±3.28	1054.40±4.18
C5	1350.18±4.77	1324.62±9.51	1320.37±6.94	1329.14±7.47	1335.34±4.50	1336.41±6.58
G18	1064.56±5.37	1043.79±4.85	1043.64±5.25	1046.77±5.12	1050.86±5.52	1052.73±5.99
G19	1461.64±7.60	1431.03±5.66	1430.41±6.44	1433.15±5.50	1437.02±6.37	1445.15±10.29
G20	1958.09±6.43	1916.35±12.62	1914.04±9.36	1917.30±7.93	1921.85±9.98	1927.90±11.38

Table 5: Obtained solution quality of 32 cooperating colonies, according to number of ant assigned per colony, where n is number of customers. Termination criterion is fixed execution time.

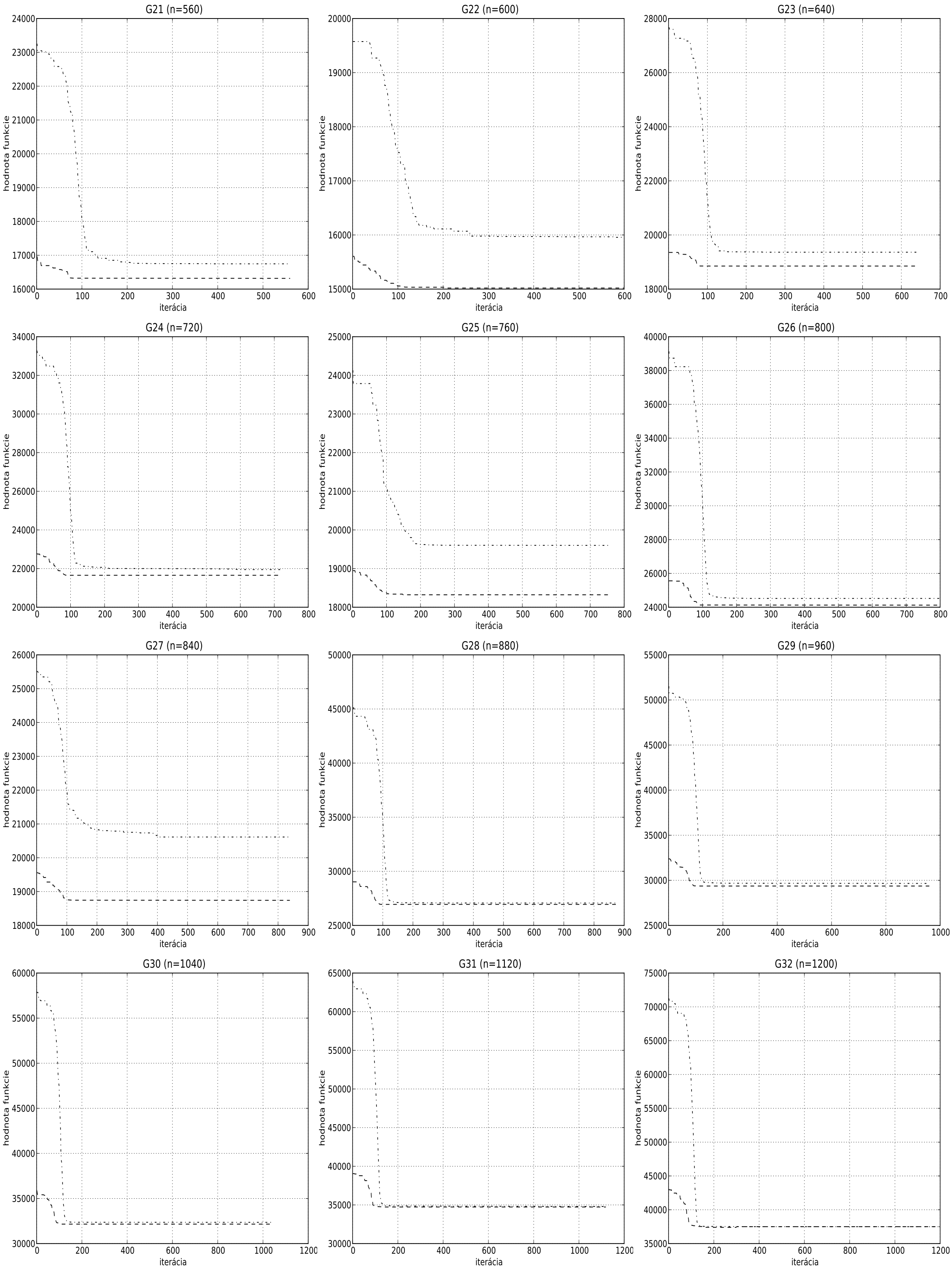

Figure 7: Iterative development of solution quality of the very large scale vehicle routing problem instances with and without using of the sweep heuristics.

instance	$\eta = 0$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$	$\eta = 0.5$	$\eta = 0.6$	$\eta = 0.7$	$\eta = 0.8$	$\eta = 0.9$
G21	16906.37	16328.27	16320.76	16320.76	16316.48	16315.02	16315.02	16429.29	16475.59
G27	20391.99	18729.37	18778.24	18751.49	18727.54	18755.98	18831.36	19487.25	20823.04
G32	38064.83	37726.72	37690.51	37372.73	37504.08	37494.65	37078.40	37327.35	37536.96

Table 6: Obtained solution qualities with different influence of the sweep heuristics η when 32 cooperating colonies were executed in parallel.

instance	Q_{avg}	GAP_{avg}	t_r[s]	t_c[ms]	Q_b	GAP_b
G21	16314.45±3.70	0.63%	556.04±0.21	19.00±1.75	16310.74	0.60%
G22	14885.98±64.80	2.07%	756.21±0.27	20.37±2.02	14808.60	1.54%
G23	18830.00±4.85	0.15%	1060.11±0.55	23.22±2.16	18825.34	0.13%
G24	21562.22±37.98	0.81%	1914.95±1.13	26.37±3.69	21508.65	0.56%
G25	18155.41±72.50	8.30%	2117.28±2.15	26.86±4.54	18038.42	7.60%
G26	24077.84±11.43	0.44%	2775.31±1.10	29.19±4.94	24059.07	0.36%
G27	18778.05±3.70	7.72%	3971.53±1.04	33.18±4.51	18636.38	6.91%
G28	26711.82±64.80	0.55%	4435.95±2.95	33.36±3.38	26674.61	0.41%
G29	29276.70±4.85	0.42%	6898.09±3.21	38.76±8.43	29229.04	0.26%
G30	31924.58±37.98	0.57%	12266.98±3.57	49.81±11.14	31892.85	0.47%
G31	34366.23±72.50	0.10%	14692.52±6.33	49.39±4.49	34349.65	0.05%
G32	37078.84±11.43	0.43%	24624.71±7.56	58.31±13.94	37060.73	0.38%

Table 7: Obtained solution quality of the very large scale vehicle routing problem running on 32 colonies, where Q_{avg} means average solution quality, GAP_{avg} shows gap to the best-know solution, t_r means running time, t_c means time spent by communication, Q_b means our best achieved solution and GAP_b means its gap to the estimated best-known solution.

hours and we get our solution in 6.8 hours. We can say that our approach is perspective for applying on the supply problems, including the variations of vehicle routing problems. According to results we can recommend using of asynchronous communication of ant colonies. Both from the quality and effectiveness point of view. It is good to apply termination criterion based on execution time. In case of small amount of used artificial ants it is appropriate to use asynchronous communication strategy which exchange pheromone matrix. On the other hand, if the amount of ants is higher it is appropriate to carry only several elite results. Further development could find such solution characteristics which identifies perspective solutions to be applied on pheromone matrix. The combination of genetic algorithms and the ant colony approach in parallel environment could be also researched. The ant colony could mutate received solution during communication. The ACOptim tool could be enlarged with computing module to solve other transport combinatorial problems. Also new local search methods for improving solution quality could be implemented.

References

[1] E. Alba and B. Dorronsoro. *Cellular genetic algorithms*, volume 42. Springer Verlag, 2008.

[2] P. R. Bell and J. E. McMullenb. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18:41–48, 2004.

[3] B. Bullnheimer, R. F. Hartl, and C. Strauss. A new rank-based version of the Ant System: A computational study. *Central European Journal of Operations Research & Economics*, 7:25–38, 1997.

[4] B. Bullnheimer, R. F. Hartl, and C. Strauss. Applying the Ant System to the vehicle routing problem. page 285–296, Netherlands, 1999. Kluwer Academic Publishers.

[5] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, M. A., T. P., and S. C., editors, *Combinatorial Optimization*, page 315–338. 1979.

[6] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–681, 1964.

[7] T. G. Crainic and M. Toulouse. Parallel Meta-heuristics. *Handbook of metaheuristics*, page 497–541, 2010.

[8] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, 1990.

[9] K. F. Doerner, R. F. Hartl, S. Benker, and M. Lucká. Cooperative Savings based ACO - Multiple Search and Decomposition Approach. *Parallel Processing Letters*, 16:351–369, 2006.

[10] K. F. Doerner, R. F. Hartl, G. Kiechle, M. Lucká, and M. Reimann. Parallel Ant Systems for the Capacitated Vehicle Routing Problem. In J. R. G. Gottlieb, editor, *Evolutionary Computation in Combinat. Optimization*, volume 3004, chapter 8, page 72–83. Springer Berlin / Heidelberg, 2004.

[11] M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical report, Italy, 1991.

[12] M. Dorigo and T. St utzle. *Ant Colony Optimization*. The MIT Press, Cambridge, Massachusetts, 2004.

[13] I. Ellabib, P. Calamai, and O. Basir. Exchange strategies for multiple ant colony system. *Information Sciences*, 177:1248–1264, 2007.

[14] B. Golden, E. Wasil, J. Kelly, and I. Chao. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. *Fleet management and logistics*, page 33–56, 1998.

[15] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[16] M. Janson, S. Merkle, and D. Middendorf. Parallel ant colony algorithms. In E. Alba, editor, *Parallel Metaheuristics: A New Class of Algorithms*, chapter 8, page 171–202. John Wiley & Sons, 2005.

[17] M. Manfrin, M. Birattari, M. St utzle, and T. Dorigo. Parallel ant colony optimization for the traveling salesman problem. In M. M. A. P. R. S. T. Birattari, editor, *Ant Colony Optimization and Swarm Intelligence*, page 224–234. Springer Berlin / Heidelberg, 2006.

[18] F. Li, B. Golden, and E. Wasil. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165–1179, 2005.

[19] M. Pedemonte, S. Nesmachnow, and H. Cancela. A survey on parallel ant colony optimization. *Applied Soft Computing*, 11(8):5181–5197, 2011.

[20] M. Reimann, M. Stummer, and K. F. Doerner. A savings based ant system for the vehicle routing problem. page 1317–1325, San Francisco, 2002. Morgan Kaufmann.

Selected Papers by the Author

M. Lucká, S. Piecka, Parallel Posix Threads Based Ant Colony Optimization using Asynchronous Communications. In *Journal of Applied Mathematics*, Volume II (2009), Number II, pp. 229-237, 2009.

M. Lucká, S. Piecka, Multi-Threaded Ant Colony Optimization with Asynchronous Communications for the Vehicle Routing Problem In *Communications*, Volume IV (2009), pp. 5-8, 2009.

S. Piecka, M. Lucká, Fine-Grained Cooperative Multi-Threaded Ant Colony Optimization with Asynchronous Communications for the Vehicle Routing Problem. In *Proceedings of 8th European Conference of Young Research and Scientific Workers TRANSCOM 2009*, University of Žilina, Žilina, pp. 217-220, 2009.

M. Lucká, S. Piecka, Ant Colony Optimizer with Application to the Vehicle Routing Problem In *Journal of Applied Mathematics*, Volume IV (2011), Number II, pp. 391-401, 2011.