ANT COLONY OPTIMIZATION WITH REINITIALISATION

M. Ciba, I. Sekaj

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava,
Ilkovičova 3, 812 19 Bratislava, Slovak Republic

Abstract

Ant colony optimization (ACO) represents an efficient tool for optimization and design of graph oriented problems. It is a multi-agent meta-heuristic approach. It is used for logistic, vehicle routing, minimal path search and many other problems which can be transformed to graph representation. The paper is devoted to extensive simulation tests and performance comparison of new developed variants based on re-initialization with the most efficient ACO variant of Kumar, et al. [2] in the Matlab environment.

1 Principles of the ACO algorithm

During the search process each ant set off from ant colony, which represents the starting position and start to move and search for food. The aim is to find the shortest way to the food location. As ants are passing the terrain (graph) they mark the used routes (arcs of the graph) by a chemical substance called pheromone. On their way back they use the same way from which abundant loops has been removed. The amount of pheromone (1) $\Delta\tau_{ij}^{k}(t)$ they produced is inversely proportional to the tour length $L^{k}(t)$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q / L^{k}(t) & if (i,j) \in \mathrm{T}^{k}(t) \\ 0 & if (i,j) \notin \mathrm{T}^{k}(t) \end{cases} \tag{1}$$

$\mathrm{T}^{k}(t)$ is the tour generated by $k-th$ ant, Q is a constant and tuple (i,j) denotes beginning and termination node of a graph arc. The amount of pheromone passed by each ant represents a quality of the particular solution. All pheromone tracks (2) are preserved by arcs of the graph

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{2}$$

where $\rho \in (0,1)$ is the pheromone persistence ($\rho - 1$ is evaporation rate) and m is the number of ants. Evaporation rate is a user adjusted parameter and affects pheromone durability; i.e. how long the acquired information will be available. Too high values causes random search, too low values get algorithm stock in local optimum.

An ant in each node has to make a decision which arc to take. At the beginning when no pheromone values are available heuristic values η_{ij} takes dominance. Later the ant uses probability selection rule to choose the next arc according to

$$p_{ij}^{k}(t) = \frac{p_{ij}(t)}{\sum_{i \in N_i^k} p_{ij}(t)} \tag{3}$$

where $p_{ij}^{k}(t)$ is probability the ant k chooses the arc (i,j) from the neighbourhood N_i^k of node i except the node visited previously. The more pheromone is located on particular arc, the more attractive it is for the ant. The probability $p_{ij}(t)$ of choosing the particular arc (i,j) depends on pheromone $\tau_{ij}(t)$ and the heuristic η_{ij} values it has associated according the equation

$$p_{ij}(t) = \frac{\tau_{ij}^{\alpha}(t) + \eta_{ij}^{\beta}}{\sum_{k \in N_i} (\tau_{ik}^{\alpha}(t) + \eta_{ik}^{\beta})} \tag{4}$$

where α and β are weight parameters which represents balance between the user preferred search area and ant's gathered knowledge. Heuristic values $\eta_{i,j}$ are significant only at the beginning of the search process when pheromone values are low. They serve as user input to navigate the search process to preferred area.

To the disadvantages of ACO algorithms belong (i) many user tuneable parameters and (ii) the selection pressure. While the first one is in nature of the algorithm, to the second one many papers have been devoted. Let's mention MAX-MIN ant system (MMAS) [3, 4] in which pheromone values are restricted to an interval $(\tau_{\min}, \tau_{\max})$; identification the most versatile values for ACO parameters [5]; a modification of Ant system (AS) called AS_{rank} [6] where only $\sigma - 1$ ants are allowed to update their pheromone track; ant colony system (ACS) with *pseudo-random proportional* rule [7] in which random uniformly distributed variable $q \in \langle 0,1 \rangle$ is compared with a tuneable parameter $q_0 \in \langle 0,1 \rangle$. If $q \le q_0$ then

$$p_{ij}^{k}(t) = \begin{cases} 1 & if\ j = \arg\max p_{ij} \\ 0 & otherwise \end{cases} \tag{5}$$

else the probability selection rule (3) is applied; random selection applied to AS_{rank} [8] where random selection rate r is the probability of random selection and it represents an user parameter which adjust balance between exploration and exploitation; prevention of quick convergence (i) and stagnation avoidance (ii) mechanisms applied to AS [2].

The prevention of quick convergence mechanism is based on *pseudo-random proportional* rule [7], but the tuneable parameter q_0 is dependent on algorithm iteration

$$q_0 = \frac{\log_e(NC)}{\log_e(N\_\max)} \tag{6}$$

where NC is the current iteration and N_max is the termination iteration.

The stagnation avoidance mechanism is based on comparison of randomly generated quantity $q \in (0,1)$ with probability $p_{ij}^{k}(t)$ of selected arc. If $q \ge p_{ij}^{k}(t)$, then choose the next node randomly. This occurs in later stages of the search process, where pheromone values tend to be high, and thus chance of further exploration is low.

2 Re-initialization approach in ACO

Re-initialization applied to ACO called ACO with macro cycles (ACO_{MC}) has been introduced in [9]. The re-initialization prevents pheromone saturation and subsequently the search process from being entrapped in local optimum. The idea is based on pheromone accumulation behaviour (Fig. 1) and its limit value is

$$\tau_{ij}(\infty) = m \frac{Q}{L^{+}(1-\rho)} \tag{7}$$

where Q is a constant from equation (1), ρ is pheromone persistence from equation (2) and L^{+} is the length of the most attractive path T^{+}. The most attractive path is the path with the highest pheromone values.

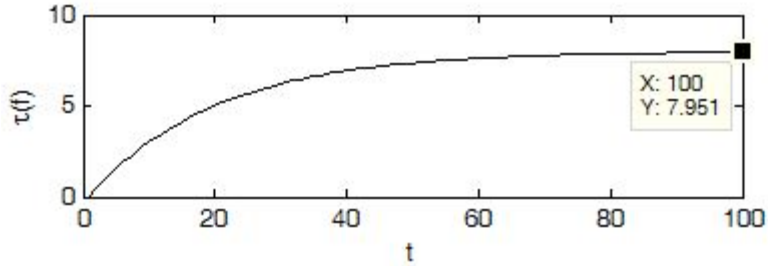

Figure 1: Simulation of pheromone accumulation for parameters $m = 2$, $Q = 1$, $L^{+} = 5$ and $\rho = 0.95$ gets limit value 8.

According to the pheromone accumulation, let us recognise three phases of search process: beginning (Figure 1, $t \in \langle 0, 40 \rangle$), saturation ($t \in \langle 40, 80 \rangle$) and stagnation ($t > 80$). The search process is re-initialized at the beginning of the saturation phase. To determine the transition between the beginning and the saturation phase equation (8) cannot be used, since the length of the most attractive path nor number of ants constituting is not known. However, derivation of the pheromone accumulation can be used instead

$$0.01\bar{\tau}(t) = \frac{d\bar{\tau}(t)}{dt}. \tag{8}$$

Low derivation values indicate the saturation phase. The transition according to the equation (7) is depicted in Fig. 2.

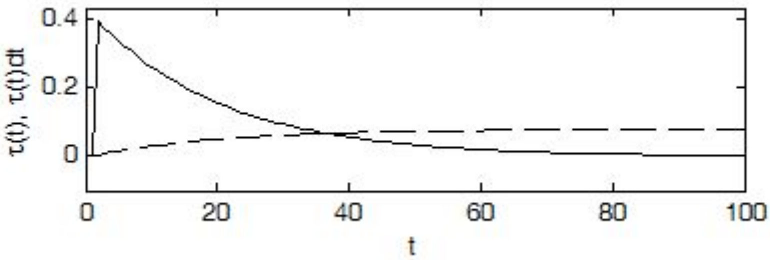

Figure 2: Bounds between beginning and the saturation phase.

Such a search process is divided into low number of macro cycles. New search process is not entirely independent from the previous one; it benefits from diminished picture of pheromone information acquired in the previous macro cycle, while the first one relies on the heuristic values provided by the user. For that purpose the projection according the following equation is used

$$f(\tau_{ij}(t)) = (s_{ij}(t)d_{ij}(t) + \bar{\tau}(t))rc \tag{9}$$

where $\bar{\tau}(t) = \sum_{ij \in G} \tau_{ij}(t)$ is mean value of the pheromone, $s_{ij}(t) = sign(\tau_{ij}(t) - \bar{\tau}(t))$ is difference sign, $d_{ij}(t) = \log_{10}\left(1 + \left|\tau_{ij}(t) - \bar{\tau}(t)\right|\right)$ is non-linear transformation of the difference and $rc = m\frac{Q}{L^{+}}$ is a reducing coefficient. The prevention of a quick convergence from Kumar, et al. (2003) is used, but value of the q_0 parameter is constant during one macro cycle. For subsequent macro cycles it is

$$q_0 \in (q_{01}, q_{02}, \ldots, q_N); \quad q_{0i=1} - q_{0i} = \frac{P_{\max} - P_{\min}}{N - 1} \tag{10}$$

where $P_{\max}$ and $P_{\min}$ corresponds to the limits of the interval $\langle 0.1, 0.9 \rangle$ respectively and N is number of macro cycles.

Further modification of ACO_{MC} is a variable value for q_0 parameter during one macro cycle. It is called ACO with variable macro cycles (ACO_{VMC}) [10]. Since the length of macro cycle varies, the mechanism capable of monitoring the search process within a single macro cycle is based on difference between mean pheromone value $\bar{\tau}(t)$ and its derivation $\frac{d\bar{\tau}(t)}{dt}$ according to equation

$$q_0(t) = 1 - k\left(\frac{d\bar{\tau}(t)}{dt} - 0.01\tau(t)\right) \tag{11}$$

where parameter $k = \frac{1}{d\bar{\tau}(t)/dt}$ insures that q_0 varies from 0 to 1.

3 Case study

The above two modifications of the ACO algorithm with re-initialisation were tested and then compared with the most efficient algorithm ACO_{KTS}. Common parameters for both algorithms were set in accordance with [5] and are listed in the Table 1.

Table 1: COMMON ACO PARAMETERS SETTINGS

Initial pheromone value $\tau_{ij}(0)$	0.1
Weight of pheromone information α	0.5
Heuristic values η_{ij}	0.1
Weight of heuristic information β	0.1
Pheromone persistence ρ	0.95

During the test available resources like number of ants or number of cycles were changed. For each setting 500 trials were performed for statistical significance. The test was performed on two different graphs.

Since the ACO_{KTS} does not use macro cycles, the first ACO versions with macro cycles were tested. Then the mean of the termination cycles from the best result was determined and considered as the input parameter for ACO_{KTS}.

3.1 Test on the 50 node graph

The first graph is a randomly generated graph with 50 nodes and 200 arcs. Node coordinates x, y are from range $\langle 0,1 \rangle$ and each arc value c_{ij} is equal to the length of the arc a_{ij}. This makes the difference between c_{ij} values very small. The graph is a symmetrical multi-graph (Fig. 3) with the depicted shortest path found during the test between start node $n_s = 2$ and end node $n_e = 31$ (green).

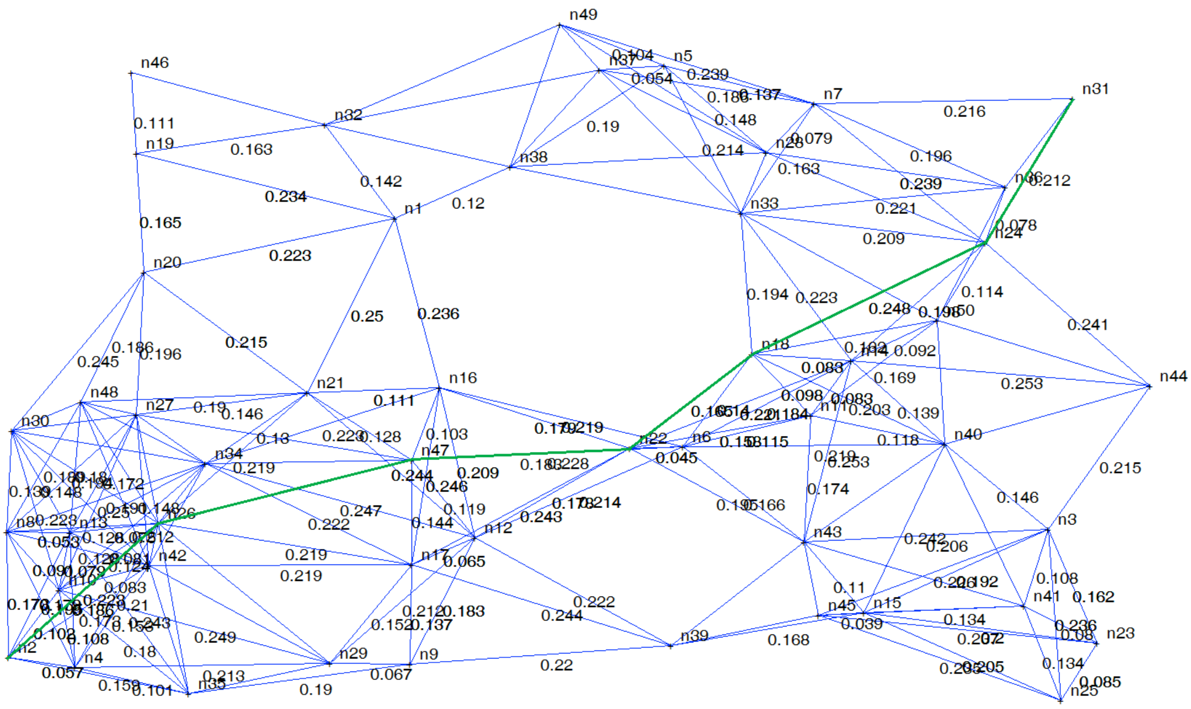

Figure 3: The 50 node graph with the minimal path - green

During the test the value of the reducing coefficient rc was changed to investigate the impact on the performance. Instead to determine its value form (9) different constant settings are used according to the Table 2.

Table 2: REDUCING COEFFICENT SETTINGS

First set	$rc =$ var iable
Second set	$rc = 0.3$
Third set	$rc = 0.2$
Fourth set	$rc = 0.1$
Fifth set	$rc = 0.05$

One test set is constituted by a couple of different groups with different available resources according to the Table 3.

Table 3: AVAILABLE RESOURCES PER GROUP

Group	Ants	Macro cycles
1	2	2
2	2	4
3	2	6
4	4	2
5	4	4
6	4	6
7	6	2
8	6	4
9	6	6

3.1.1 Minimal path search results

For test evaluation the following metrics were used. The cardinality n is the number of occurrences of the global optimum in 500 repetitions. $\bar{c}_{best}$ is the arithmetic mean of the cycle numbers where the algorithm has found the best solution and $\bar{c}_{ter}$ is the arithmetic mean of the termination cycle numbers.

During the test all algorithms and variants with different settings were able to find the global optimum given by the minimal path $T_{\min} = \{2\ 26\ 47\ 22\ 18\ 24\ 31\}$ and its length $L_{\min} = 1.26023855725949$. The only difference was in cardinality n and the mean value of the best cycle number $\bar{c}_{best}$ and of the termination cycle number $\bar{c}_{ter}$.

The results of minimal path search show Tables 4 – 14. There is a separate table for each set where each line represents statistic values from 500 trials.

Table 4: ACO_{VMC} for rc = var

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	19	23.421	40.895
2	52	42.846	81.365
3	83	72.554	125.000
4	39	23.538	44.590
5	93	47.849	90.516
6	141	70.837	137.050
7	75	24.227	45.293
8	129	48.442	92.713
9	175	67.994	140.491

Table 5: ACO_{MC} for rc = var

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	34	23.147	42.941
2	63	54.349	86.651
3	102	81.716	129.755
4	57	24.982	43.368
5	119	49.395	91.555
6	173	76.532	137.491
7	72	25.472	46.458
8	141	51.858	93.759
9	185	75.005	140.589

Table 6: ACO_{VMC} for rc = 0.3

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	15	20.533	33.267
2	46	33.391	65.652
3	57	55.965	95.737
4	39	23.590	41.282
5	83	41.434	82.554
6	100	64.210	122.480
7	59	23.695	44.441
8	115	43.739	86.652
9	163	63.080	131.025

Table 7: ACO_{MC} for rc = 0.3

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	14	26.143	35.143
2	38	39.868	66.789
3	53	52.038	98.189
4	48	21.208	41.896
5	72	48.722	82.139
6	127	65.291	122.772
7	49	22.776	44.531
8	115	46.348	87.757
9	143	69.203	131.860

Table 8: ACO_{VMC} for rc = 0.2

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	20	20.200	36.750
2	40	39.325	70.975
3	83	61.181	107.145
4	45	19.022	42.622
5	85	41.318	85.212
6	116	66.767	127.328
7	62	23.839	44.984
8	127	47.228	90.094
9	178	65.534	133.522

Table 9: ACO_{MC} for rc = 0.2

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	23	23.391	38.478
2	39	44.897	72.308
3	60	62.167	108.317
4	49	25.041	43.000
5	101	50.683	85.485
6	128	61.398	127.953
7	67	24.687	44.179
8	130	50.554	89.346
9	176	69.028	135.477

Table 10: ACO_{VMC} for rc = 0.1

Group	n	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	19	28.158	40.105
2	56	48.125	77.929
3	69	63.899	116.739
4	44	25.864	43.545
5	93	41.925	87.667
6	132	66.992	132.720
7	63	20.889	44.841
8	136	45.184	89.662
9	167	63.784	137.198

Table 11: ACO_{MC} for rc = 0.1

Group	n	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	29	25.345	40.207
2	49	50.959	80.061
3	76	66.539	117.711
4	49	28.082	43.776
5	96	48.594	88.260
6	132	79.174	131.939
7	52	24.481	45.673
8	133	48.820	92.774
9	164	74.799	137.567

Table 12: ACO_{VMC} for rc = 0.05

Group	n	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	29	26.552	39.034
2	53	52.434	81.377
3	77	68.351	120.117
4	47	26.979	44.638
5	83	47.325	89.277
6	114	61.719	134.289
7	82	26.598	45.659
8	134	45.358	91.888
9	172	64.291	138.238

Table 13: ACO_{MC} for rc = 0.05

Group	n	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	26	27.346	41.077
2	60	48.567	83.867
3	82	77.305	121.659
4	52	23.365	43.962
5	103	51.495	90.738
6	152	77.401	134.796
7	60	27.283	45.300
8	130	54.977	92.215
9	205	71.473	139.054

Table 14: ACO_{KTS}

Group	n	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	29	26.552	43
2	70	45.214	87
3	117	70.829	130
4	66	26.152	43
5	113	47.814	92
6	200	72.320	137
7	100	26.430	47
8	192	45.406	94
9	252	66.313	141

Both ACO variants with macro cycles achieved the best performance with reducing coefficient which was set to variable. The simple variant ACO_{MC} outperformed ACO_{VMC} in the most cases.

The comparison of ACO_{KTS} with best delivered results from macro cycle variant (ACO_{MC}, Table 5) reveals ACO_{KTS} superior performance in each group (Table 14). The differences become larger with more resources available.

3.1.2 Results for the maximal path search

The graph, base parameters and reducing coefficient setting are the same as for the minimal path search case. Available resources, like number of ants and macro cycles, are in the Table 15. In each trial a different local optimum was found. Local optimum is given by the path $T_{\max}$ and it's length $L_{\max}$. Since each settings with 500 trials found different local optimum, results in Tables 16 – 26 are extended by the length of the best solution $L_{\max}$ and arithmetic mean of all $L_{\max}$ values $\overline{L}_{\max}$.

Table 15: AVAILABLE RESOURCES PER GROUP

Group	Ants	Macro cycles
1	4	4
2	4	6
3	4	8
4	6	4
5	6	6
6	6	8
7	8	4
8	8	6
9	8	8

Table 16: ACO_{VMC} for rc = var

Group	$L_{\max}$	$\overline{L}_{\max}$	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	6.044	3.905	44.064	89.922
2	5.330	4.042	65.290	134.898
3	5.332	4.113	88.410	180.306
4	5.445	4.047	45.340	92.352
5	5.319	4.091	68.822	139.052
6	5.533	4.222	89.410	186.324
7	5.578	4.094	46.088	92.160
8	5.446	4.206	70.552	140.190
9	5.309	4.270	88.480	187.490

Table 17: ACO_{MC} for rc = var

Group	$L_{\max}$	$\overline{L}_{\max}$	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	5.081	3.913	34.460	68.548
2	5.205	3.981	51.022	102.714
3	5.796	4.082	70.002	138.168
4	5.363	4.001	35.718	70.624
5	5.253	4.121	52.794	107.886
6	5.542	4.210	74.350	143.352
7	5.678	4.102	36.702	72.858
8	5.309	4.194	55.532	110.012
9	5.504	4.270	72.374	147.498

Table 18: ACO_{VMC} for rc = 0.3

Group	$L_{\max}$	$\overline{L}_{\max}$	$\overline{c}_{best}$	$\overline{c}_{ter}$
4.914	3.892	38.572	81.332	4.914
5.499	4.007	60.842	121.166	5.499
5.416	4.102	75.902	161.314	5.416
5.063	4.019	43.416	86.900	5.063
5.516	4.110	65.134	129.760	5.516
5.187	4.177	84.870	173.012	5.187
5.441	4.101	42.590	88.908	5.441
5.814	4.177	65.520	133.604	5.814
5.445	4.246	86.104	178.298	5.445

Table 19: ACO_{MC} for rc = 0.3

Group	$L_{\max}$	$\overline{L}_{\max}$	$\overline{c}_{best}$	$\overline{c}_{ter}$
1	4.883	3.880	32.856	67.448
2	5.138	3.998	49.892	99.914
3	5.674	4.061	66.734	133.254
4	5.541	4.041	35.604	70.798
5	5.402	4.083	53.282	105.564
6	5.243	4.192	72.992	142.160
7	5.524	4.078	36.018	72.574
8	5.384	4.194	57.322	110.048
9	5.695	4.277	72.950	148.266

Table 20: ACO_{VMC} for rc = 0.2

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
5.142	3.900	41.104	84.540	5.142
5.397	4.024	61.982	125.892	5.397
5.581	4.098	81.950	167.378	5.581
5.537	3.997	42.398	88.696	5.537
5.618	4.121	64.878	133.220	5.618
5.532	4.178	86.460	177.428	5.532
5.387	4.080	43.748	89.924	5.387
5.423	4.209	65.272	135.436	5.423
5.726	4.258	92.486	181.210	5.726

Table 21: ACO_{MC} for rc = 0.2

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
5.433	3.885	33.524	67.392	5.433
5.569	4.024	51.644	101.748	5.569
5.201	4.077	65.446	134.454	5.201
5.360	3.996	35.710	71.174	5.360
5.765	4.127	52.264	106.570	5.765
5.680	4.200	68.572	143.542	5.680
5.457	4.099	37.780	73.356	5.457
5.540	4.176	53.892	111.016	5.540
5.550	4.297	73.896	148.392	5.550

Table 22: ACO_{VMC} for rc = 0.1

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	5.174	3.921	44.260	86.946
2	5.305	4.015	63.664	130.738
3	5.604	4.097	83.580	173.886
4	5.362	4.033	43.924	90.828
5	5.380	4.137	67.598	135.826
6	5.435	4.195	92.174	182.040
7	5.181	4.078	45.776	92.204
8	5.336	4.202	68.340	138.040
9	5.621	4.246	87.920	184.628

Table 23: ACO_{MC} for rc = 0.1

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	6.208	3.879	33.124	68.402
2	5.271	3.999	48.930	101.882
3	5.597	4.091	67.714	135.590
4	5.124	4.016	35.190	71.502
5	5.534	4.143	53.434	107.900
6	5.645	4.185	69.630	143.328
7	5.523	4.101	37.048	72.584
8	5.466	4.192	55.104	110.972
9	5.637	4.284	77.918	149.960

Table 24: ACO_{VMC} for rc = 0.05

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	5.372	3.947	42.400	88.362
2	5.178	4.015	69.290	133.076
3	5.807	4.124	90.172	177.030
4	5.307	3.997	43.308	91.968
5	5.393	4.133	66.812	137.390
6	5.723	4.216	90.742	183.598
7	5.399	4.081	45.412	92.672
8	5.381	4.199	68.016	139.458
9	5.513	4.289	90.216	185.814

Table 25: ACO_{MC} for rc = 0.05

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	5.267	3.885	35.314	68.600
2	5.752	4.028	49.984	102.628
3	5.312	4.058	69.240	136.998
4	6.079	4.009	34.152	71.746
5	6.005	4.130	54.092	108.154
6	5.435	4.214	70.286	145.074
7	5.378	4.096	37.278	73.908
8	5.453	4.210	56.564	111.952
9	5.493	4.273	72.212	149.908

Table 26: ACO_{KTS}

Group	$L_{\max}$	$\bar{L}_{\max}$	$\bar{c}_{best}$	c_{ter}
1	5.091	3.966	42.768	90.000
2	5.253	4.058	65.470	135.000
3	5.299	4.130	84.088	180.000
4	5.211	4.058	43.656	92.000
5	5.538	4.171	64.506	139.000
6	5.974	4.235	92.780	186.000
7	5.937	4.131	41.924	92.000
8	5.710	4.221	68.200	140.000
9	5.595	4.332	88.494	187.000

ACO_{VMC} has a longer search process in each set, but the highest best value $\overline{L}_{\max}$ varies between ACO_{VMC} and ACO_{MC}. Constant values for the reducing coefficient cause decrease in performance. Better results were achieved with lower values of the reducing coefficient. It could even outperform variable settings with 0.05 for ACO_{VMC} and with 0.02 for ACO_{MC}. Decreasing of the values of the reducing coefficient causes longer search process for ACO_{VMC}, but it did not have the same impact on ACO_{MC}.

The mean values for the termination cycle were taken form ACO_{VMC} (Table 16) and were used for ACO_{KTS} (Table 26). The results reveal ACO_{KTS} outperforms ACO_{VMC} in terms of $\overline{L}_{\max}$ in each group. However, the difference is very small.

3.2 Test on the 62 node graph

The second graph is an asymmetric multi graph with 62 nodes and 114 arcs in which ants are allowed to take any arc in direction from the left to the right only. The start node is $n_s = 62$ and the end node is $n_e = 21$. Figure 4 shows the graph with the best path in red.

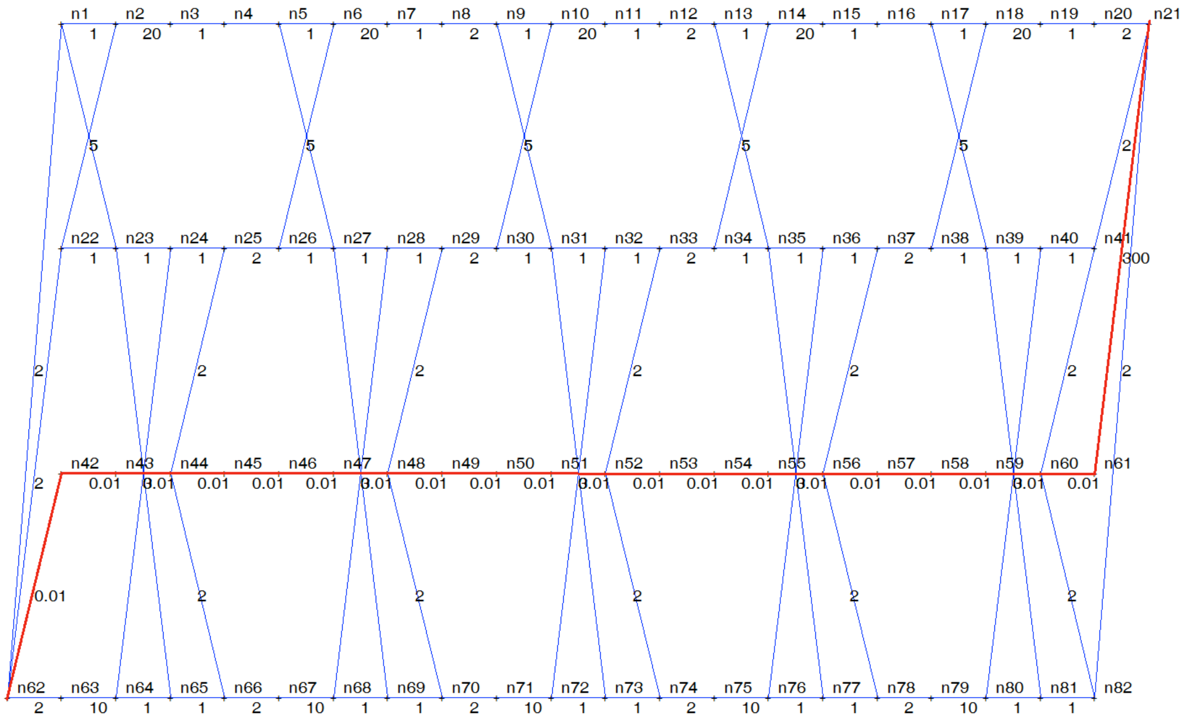

Figure 4: The 62 node graph with maximal path - red

The resources represented by number of ants and number of macro cycles are in Table 3. The reducing coefficient was set as variable. Notice that this graph was designed to lead the greedy algorithms out of the global optimum. Ant need to take arcs with low values and at the end it receives 300 on the last arc $a_{61,21}$.

All algorithms with all different parameter settings were able to find the longest paths {62 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 21} with value $L^+ = 300.2$. The only difference was in cardinality n and arithmetic mean of the best value cycles $\overline{c}_{best}$. As the test results show, there is a difference between ACO_{VMC} and ACO_{MC} (Table 27 and 29) performance. Thus test for ACO_{KTS} was running twice; once for termination cycles set according to ACO_{VMC} (Table 28) and once according to ACO_{MC} (Table 30). That allowed the performance cross comparison between both versions of ACO with macro cycles and ACO_{KTS}.

Table 27: ACO_{VMC}

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	11	6.364	10.545
2	14	11.857	29.071
3	28	20.107	44.536
4	15	5.867	14.067
5	53	14.151	37.434
6	54	22.259	59.204
7	43	6.186	18.698
8	80	18.138	45.313
9	132	31.394	71.485

Table 28: ACO_{KTS}

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	4	7.000	11
2	15	10.133	29
3	13	13.846	45
4	13	4.462	14
5	24	9.417	37
6	25	15.360	59
7	26	6.000	19
8	38	10.474	45
9	37	17.583	71

Table 29: ACO_{MC}

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	14	10.857	23.643
2	37	19.703	50.622
3	59	37.898	79.678
4	33	13.091	21.212
5	41	12.634	25.146
6	46	15.326	26.957
7	41	8.829	23.244
8	52	9.981	24.327
9	42	13.286	25.714

Table 30: ACO_{KTS}

Group	n	$\bar{c}_{best}$	$\bar{c}_{ter}$
1	14	7.214	24
2	17	9.765	51
3	27	15.259	80
4	17	6.235	21
5	17	11.118	25
6	20	6.450	27
7	23	8.087	23
8	30	4.467	24
9	22	6.636	26

ACO_{MC} leads to higher cardinality and longer search process only for limited resources, i.e. up to 6 ants and 4 macro cycles (Table 27, 29). Then ACO_{VMC} higher cardinality and longer search process. The more resources are available the bigger the difference is.

The performance of ACO_{KTS} is worse than any ACO variant with macro cycle, especially ACO_{VMC} (Table 27, 28). The difference only increases with increasing number of resources.

4 Conclusion

In the 50 node graph path minimisation, ACO_{KTS} outperforms the ACO_{MC}. However, for maximal path search the difference between ACO_{KTS} and ACO_{VMC} is very small. In the 62 node graph, where the success depends on finding a single well hidden global optimum, both variants ACO with macro cycles outperform ACO_{KTS}.

In the 50 node graph, ACO_{MC} outperforms ACO_{VMC} for minimal path search. For maximal path search the best result varies between ACO_{MC} and ACO_{VMC}. In the 62 node graph, ACO_{MC} outperforms ACO_{VMC} only with the use of limited resources. With more resources, ACO_{VMC} outperforms ACO_{MC} and the difference increases with increasing number of resources.

The influence of the reducing coefficient is significant. In general, constant values cause performance decrease. The lower the constant value is the better the performance is. However, only for maximal path search the low constant values are able to outperform results obtained with variable reducing coefficient.

In general, the here obtained results are promising. They show a performance potential compared to ACO_{KTS}. Further improvement of ACO macro cycles variant will be achieved by longer search process. But the search process will have longer macro cycle instead of more macro cycles, which will allow better utilisation of already gathered knowledge. This will be obtained by re-initialisation shift to a later stage, i.e. between the saturation and stagnation phase. Further tests are necessary to evaluate the impact of such modifications. However, the benefit of more intuitive input in number of macro cycles versus number of cycles can prevent from unnecessary long search process.

References

[1] M. Dorigo, G. Caro and L. Gambardella. *Ant algorithms for discrete optimisation.* Artificial Life, MIT Press, 1991, 1-36

[2] R. Kumar, M. K. Tiwari and R. Shankar. *Scheduling of flexible manufacturing systems: an ant colony optimisation approach.* Proc. Instn. Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture, 2003, 1443–1453

[3] T. Stützle and H. Hoos. *Improvements on the ant system: Introducing MAX–MIN ant system.* In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pages 245–249. Springer Verlag, Wien, 1997.

[4] T. Stützle and H. Hoos. *The MAX–MIN ant system and local search for the travelling salesman problem.* In T. Baeck, Z. Michalewicz, and X. Yao, editors, Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, pages 309–314. IEEE Press, 1997.

[5] M. Becker and H. Szczerbicka. *Parameters influencing the performance of ant algorithms applied to optimisation of buffer size in manufacturing.* IEMS Vol. 4, No. 2, December 2005, 184–191

[6] B. Bullnheimer, R. Hartl and C. A. Straus. *A new rank-based version of the ant system: a computational study*. Technical Report POM-03*/97,* Institute of Management Science, University of Vienna, 1997. Accepted for publication in the Central European Journal for Operations Research and Economics.

[7] L. M. Gambardella and M. Dorigo. *Solving symmetric and asymmetric TSPs by ant colonies.* In Proceedings of the IEEE Conference on Evolutionary Computation, ICEC96, pages 622–627. IEEE Press, 1996.

[8] Y. Nakamichi and T. Arita. *Diversity control in ant colony optimization.* In Abbas HA (ed) Proceedings of the Inaugural Workshop on Artificial Life (AL'01), Adelaide, Australia, Dec 11, 2001, pp 70-78

[9] M. Ciba. *ACO algorithm with reinitialisation.* 14th Conference of Doctoral Students ELITECH'12, FEI SUT Bratislava, Slovakia, May 2012

[10] M. Ciba and I. Sekaj. *ACO algorithm with variable macro cycles*. 18th International Conference on Soft Computing MENDEL'12, Brno, Czech Republic, June 2012

Matej Ciba
bigmato@centrum.sk

Ivan Sekaj
ivan.sekaj@stuba.sk