A Hybrid Genetic-Ant Colony Algorithm for Travelling Salesman Problem

Emel Soylu*‡, Ali Uysal**

*Department of Mechatronics Engineering, Faculty of Technology, Karabük University, 78050, Karabük/TURKEY

** Department of Mechatronics Engineering, Faculty of Technology, Karabük University, 78050, Karabük/TURKEY

(ekocak@karabuk.edu.tr, aliuysal@karabuk.edu.tr)

‡ Corresponding Author; Emel Soylu, Department of Mechatronics Engineering, Karabuk University, Turkey, Tel: +90 370 433 82 02, Fax: +90 370 433 82 10, ekocak@karabuk.edu.tr

Received: 10.07.2017 Accepted:25.09.2017

Abstract- Travelling salesman problem is a well-known problem in optimization algorithms. In this study, we propose a hybrid genetic-ant colony algorithm to solve this problem. There are no certain formulas to determine the parameters of ant colony algorithm. Usually, programmers use the trial and error method to find best values. We use the genetic algorithm to optimize best parameter values of ant colony algorithm. In this way, the success rate of ant colony algorithm is maximized. The success of the method has been proven by experimental studies.

Keywords Ant colony algorithm; Genetic algorithm; Path planning; Hybrid genetic-ant colony algorithm.

1. Introduction

Optimization is everywhere in the life. Not only people but also other living things make an optimization to make life easier. In airline scheduling, finance, internet routing, navigation, robotic path planning and etc. Optimization is used almost all applications in engineering and industry. We optimize something to minimize the cost and energy consumption or to maximize the efficiency, performance, and profit etc. Optimization is very important in applications because energy sources, money, time is always limited [1].

Many of the algorithms used in optimization have been developed from nature. Genetic algorithm (GA) [2], particle swarm optimization [3], ant colony optimization (ACO) [4], monkey Search [5], wolf pack search algorithm [6], cuckoo search [7], fruit fly optimization algorithm [8], dolphin echolocation [9], Whale optimization algorithm [10] are some of the algorithms. Finding the best solution is the common goal of these algorithms. This can sometimes be found in an equation to find the most suitable parameters, to find the most suitable coefficients, to find the best shortest path, to make the least costly choice.

Travelling salesman problem (TSP) is a well-known problem for optimization algorithms. TSP asks for the most efficient trajectory possible given a set of nodes and distances that must all be visited. In computer science, the problem can be applied to the most efficient route for data to travel between various nodes. The goal is to find the shortest way. Ant colony optimization algorithm (ACO) is used in path planning problems [11], [12]. Genetic algorithm (GA) is frequently used in all kinds of optimization problems [13]. Path planning is one of these problems [14]–[16].

The ACO gives successful solving for TSP if the parameters are convenient. However, the efficiency of the ACO is closely related to the chosen parameters that include the information heuristic factor α, the expectation heuristic factor β, the pheromone evaporation factor ρ. Usually, programmers determine the parameters of ACO via trial-error method because there are no certain formulas to determine these values. Parameter selection differs for different types of optimization problems. Moreover, parameters may change according to the varied situations of the problem [17]. For TSP constant adjusted parameters may not be suitable when the position of the nodes in TSP problem changes. In other words, these parameters differ according to the node locations in the TSP problem. Optimization algorithms give good results for parameter estimation problems [18]. We propose a hybrid GA-ACO algorithm to overcome this problem. Firstly we determine the parameters of ACO via GA and then run ACO with these values. The efficiency of the algorithm is maximized. A graphical user interface is designed to run proposed method. The number of nodes and location of the nodes of TSP are defined by the user in that software. The difference between manual parameter adjustment and optimization of parameters are compared.

The structure of the paper is introduced as follows. Section II gives out the methods GA , ACO, and GA-ACO. In Section III, some experiments are given and the analysis of experimental results are done. Finally, some conclusion is given in Section IV.

2. Methods

GA and ACO methods are frequently used methods in optimization problems. In this section, small explanation about GA and ACO is given.

2.1. Genetic Algorithm

In programming, a genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. Genetic algorithms are usually used to obtain high-quality solutions to optimization and search problems by relying on biological operators such as mutation, crossover, and selection [2].

Basic process steps of the genetic algorithm are given below:

Step 1: Start

Step 2: Set genetic algorithm parameters

Step 3: Generate initial random population

Step 4: Evaluate fitness for each chromosome in the population

Step 5: Are optimization criteria met? If yes go to step 11 if no go to step 6

Step 6: Parents selection for next chromosome

Step 7: Crossover of parents chromosome

Step 8: Mutation of chromosome

Step 9: New population

Step 10: Go to step 4

Step 11: Best chromosome

Step 12: End [19]:

Binary numbers are used to generate chromosomes.GA is implemented with small population size to allow the controller as fast as possible. In this study, the size of the initial population is adjusted to 5, crossover rate is 5%, the mutation rate is 5%, and the number of iterations is 5.These values can be adjusted according to the number of nodes in the TSP. For a small number of nodes, small iteration number and population number will be enough nevertheless it is necessary to increase these values as the number of nodes increases for better results. The initial population is set by encoding the ACO parameters α, β, and ρ into binary strings known as a chromosome. The length of the chromosome for precision is 3 determined as following equations:

$$2^{mj-1} < (bj - aj)\cdot 10^3 \le 2^{mj} - 1 \qquad (1)$$

where mj is the number of bits, and bj and aj are upper and lower bounds of ACO parameters. For example, if $\alpha \in [0,5]$ $\beta \in [0,2]$, and $\rho \in [0,1]$, the required bits calculated according to Eq. (1) are equal to 13,11,10 respectively. The length od the chromosome is 34. The fitness of each chromosome is evaluated by converting the binary string into real value. The converting process is done according to Eq. (2) [17].

$$xj = aj + \text{decimal(subsitringj)} \cdot \frac{bj - aj}{2^{mj} - 1} \qquad (2)$$

A single crossover point on both parents' strings is selected. Elitism method is used in the selection step. This method first copies the best chromosome (or a few best chromosomes) to newly generated population. The rest is done in a classical way. Elitism can very rapidly increase the performance of GA because it prevents losing the best-found solution.

2.2. *Ant Colony Algorithm*

Ant colony optimization algorithm is introduced by Dorigo [4]. This algorithm is inspired by ants. While the ant colony is looking for the food source, they will leave pheromone to attract and guide other ants on the path that they have passed, the intensity of pheromone is in inverse proportion to the length of the path. Ants can sense the pheromone and they chose the path which has the maximum pheromone intensity. Basic process steps of the ACO are given below:

Step 1: Start

Step 2: Create ants

Step 3: Put ants on an entry state

Step 4: Create empty path lists for each ant

Step 5: Select next state for each ant

Step 6: Are the path lists full if no go to step 5 if yes go to step 7

Step 7: Update local pheromone

Step 8: Update global pheromone

Step 9: Evaporate pheromone

Step 10: Is termination criteria met? If no go to step 3 if yes go to step 11

Step 11: End

At time t, the transition probability of moving from node i to j for ant k is given in Eq. (3).

$$p_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}{\sum\limits_{u \in j_k(i)} [\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta} & \text{if } \; j \in j_k(i) \\ 0 & \text{other situations} \end{cases} \qquad (3)$$

In this equation, jk(i) is the set of accessible neighbour nodes of node i, η is the local visibility, τ is the pheromone intensity and β are the constant parameters that determine the relative of ant. After the ants in the algorithm ended their tours, the pheromone trail amount will update according to the following formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t+1) \qquad (4)$$

ρ ($0 \le \rho \le 1$) is the pheromone evaporation rate in Eq.(4)., and $\tau_{ij}(t)$ is the amount of pheromone accumulated until iteration t, $\Delta\tau_{ij}^{k}(t+1)$ is the pheromone amount at iteration t. The local pheromone amount $\Delta\tau_{ij}^{k}(t+1)$ is estimated according to Eq. 5.

$$\Delta\tau_{ij}^{k}(t+1) \begin{cases} \dfrac{1}{L^{k}(t+1)} & k^{th} \text{ ant used the node } (i,j) \\ 0 & \text{other conditions} \end{cases} \qquad (5)$$

In Eq. (5). Lk(t+1) is the total distance of path for kth ant. Local pheromone updating makes the transition paths attractive by dynamically changing tours. The ants also change their tours at every iteration depending on the changing pheromone amounts. Thus, it is always aimed to find shorter tours. The global pheromone update is calculated using a formula similar to the local pheromone update: After all ants finish their tours the best path's pheromone amount is updated according to Eq. (6). Lbest is the distance of shortest path in tour t.

$$\Delta\tau_{ij}^{k}(t+1) \begin{cases} \dfrac{1}{L_{best}(t+1)} & \text{if } (i,j) \text{ is the best tour} \\ 0 & \text{other conditions} \end{cases} \qquad (6)$$

2.3. Hybrid GA-ACO Algorithm

The block diagram of GA-ACO is given in Fig. 1. The genetic algorithm optimizes the parameters of ACO according to the output of ACO. ACO is also the fitness function of GA. ACO operates with optimized parameters and for TSP problem a path plan and the distance of the path is obtained. The distance parameter in the input value of GA. With this method, It is not necessary for the user to find the correct values by trial and error method. Basic process steps of the GA-ACO are given below:

Step 1: Start

Step 2: Set genetic algorithm parameters

Step 3: Generate initial random population

Step 4: Evaluate fitness for each chromosome in the population

Step 5: Start ACO

Step 6: Create ants

Step 7: Put ants on an entry state

Step 8: Create empty path lists for each ant

Step 9: Select next state for each ant

Step 10: Are the path lists full if no go to step 9 if yes go to step 11

Step 11: Update local pheromone

Step 12: Update global pheromone

Step 13: Evaporate pheromone

Step 14: Is termination criteria met? If no go to step 7 if yes go to step 15

Step 15: End ACO

Step 16: Are optimization criteria met? If yes go to step 22 if no go to step 17

Step 17: Parents selection for next chromosome

Step 18: Crossover of parents chromosome

Step 19: Mutation of chromosome

Step 20: New population

Step 21: Go to step 4

Step 22: Best chromosome

Step 23: End

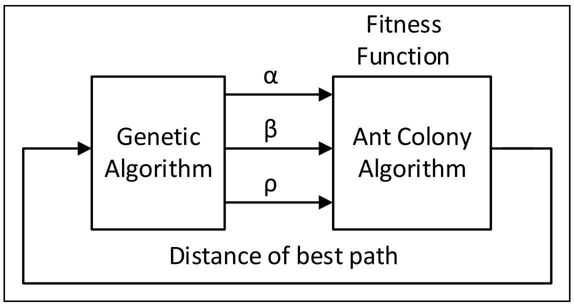


Fig. 1. Block diagram of the proposed method.

3. Experimental Study

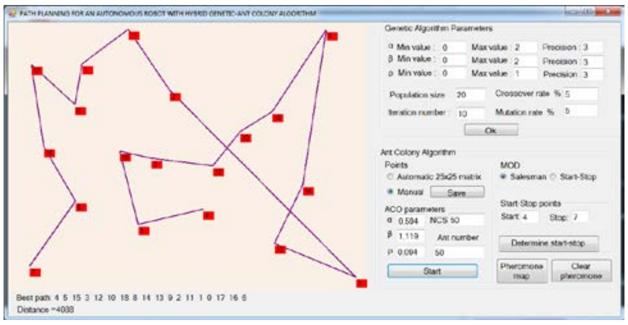
Several experiments were performed for different numbers of nodes positioned in different places in the simulation. For each experiment different ACO parameters are obtained via GA. A few of simulations are given below.

For simulation given in Fig. 2 the path map is obtained as 4, 5, 15, 3, 12, 10, 18, 8, 8, 14, 13, 9, 2, 11, 1, 0, 17, 16, 6. The distance is 4088. The parameters of α ϵ [0,2] β ϵ [0,2], and ρ ϵ [0,1] are obtained as 0.594, 1.119, 0.094 respectively. The population size is 20, iteration number is 10, crossover rate is 5%, the mutation rate is 5%. The coordinates of the simulations are given in Table 1.

For simulation given in Fig. 3. the path map is obtained as 4, 5, 14, 10, 11, 9, 13, 8, 7, 12, 6, 3, 2, 1. The distance is 1583. The parameters of α ϵ [0,1] β ϵ [0,1], and ρ ϵ [0,1] are obtained as 0.032, 0.541, 0.024 respectively. The population size is 30, iteration number is 5, crossover rate is 5%, the mutation rate is 5%. The coordinates of the simulations are given in Table 2.

Table 1. Coordinates of nodes for Fig. 2.

Order	**X**	**Y**	**Order**	**X**	**Y**	**Order**	**X**	**Y**
0	123	157	**7**	40	455	**14**	541	305
1	136	82	**8**	584	19	**15**	205	243
2	297	131	**9**	638	475	**16**	66	235
3	263	256	**10**	426	207	**17**	43	83
4	358	350	**11**	221	18	**18**	485	170
5	239	378	**12**	376	270			
6	124	335	**13**	489	413			


PATH PLANNING FOR AN AUTONOMOUS ROBOT WITH HYBRID GENETIC-ANT COLONY ALGORITHM

Genetic Algorithm Parameters
α Min value : 0 Max value : 2 Precision : 3
β Min value : 0 Max value : 2 Precision : 3
ρ Min value : 0 Max value : 1 Precision : 3
Population size : 20 Crossover rate % 5
Iteration number : 10 Mutation rate % 5

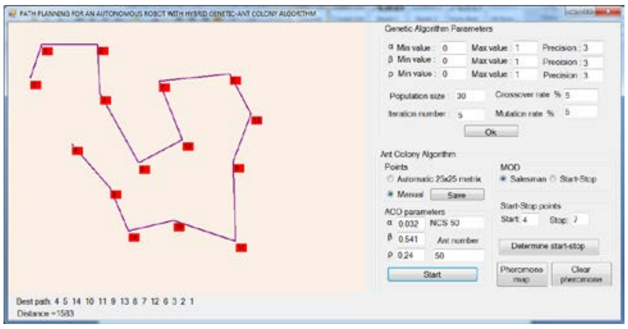
Ant Colony Algorithm
Points: ○ Automatic 25x25 matrix ● Manual
MOD: ● Salesman ○ Start-Stop
ACO parameters: α 0.584 NCS 50; β 1.119 Ant number 50; ρ 0.094
Start-Stop points: Start 4 Stop 7

Best path: 4 5 15 3 12 10 18 8 14 13 9 2 11 1 0 17 16 6
Distance =4088

Figure 2. Simulation 1.

Table 2. Coordinates of nodes for Fig. 3.

Order	**X**	**Y**	**Order**	**X**	**Y**
0	41	107	**8**	407	99
1	62	46	**9**	413	259
2	164	45	**10**	304	366
3	169	135	**11**	418	404
4	117	227	**12**	320	219
5	188	307	**13**	446	172
6	240	261	**14**	221	385
7	277	112			


PATH PLANNING FOR AN AUTONOMOUS ROBOT WITH HYBRID GENETIC-ANT COLONY ALGORITHM

Genetic Algorithm Parameters
α Min value : 0 Max value : 1 Precision : 3
β Min value : 0 Max value : 1 Precision : 3
ρ Min value : 0 Max value : 1 Precision : 3
Population size : 30 Crossover rate % 5
Iteration number : 5 Mutation rate % 5

Ant Colony Algorithm
Points: ○ Automatic 25x25 matrix ● Manual
MOD: ● Salesman ○ Start-Stop
ACO parameters: α 0.032 NCS 50; β 0.541 Ant number 50; ρ 0.24
Start-Stop points: Start 4 Stop 7

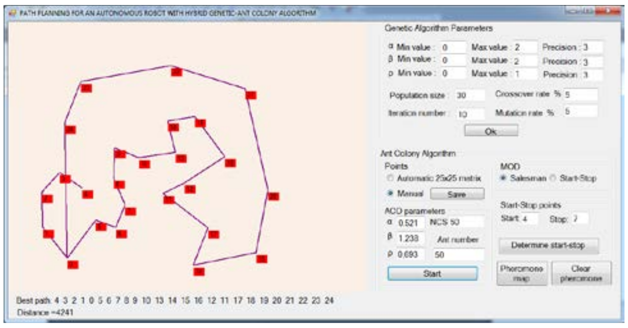
Best path: 4 5 14 10 11 9 13 8 7 12 6 3 2 1
Distance =1583

Figure 3. Simulation 2.

For simulation given in Fig. 4. the path map is obtained as 4, 3, 2, 1, 0, 5, 6, 7, 8, 9, 10, 13, 14, 15, 16, 12, 11, 17, 18, 19, 20, 21, 22, 23, 24. The distance is 4241. The parameters of α ϵ [0,2] β ϵ [0,2], and ρ ϵ [0,1] are obtained as 0.521, 1.238, 0.693 respectively. The population size is 30, iteration number is 10, crossover rate is 5%, the mutation rate is 5%. The coordinates of the simulations are given in Table 3.

Table 3. Coordinates of nodes for Fig. 4.

Order	**X**	**Y**	**Order**	**X**	**Y**	**Order**	**X**	**Y**
0	109	436	**9**	196	235	**18**	340	449
1	64	380	**10**	240	253	**19**	455	426
2	62	316	**11**	285	318	**20**	477	312
3	95	280	**12**	324	299	**21**	435	127
4	136	308	**13**	307	243	**22**	299	85
5	161	367	**14**	293	186	**23**	134	114
6	198	380	**15**	343	178	**24**	104	190
7	215	339	**16**	398	251			
8	194	290	**17**	367	381			


PATH PLANNING FOR AN AUTONOMOUS ROBOT WITH HYBRID GENETIC-ANT COLONY ALGORITHM

Genetic Algorithm Parameters
α Min value : 0 Max value : 2 Precision : 3
β Min value : 0 Max value : 2 Precision : 3
ρ Min value : 0 Max value : 1 Precision : 3
Population size : 30 Crossover rate % 5
Iteration number : 10 Mutation rate % 5

Ant Colony Algorithm
Points: ○ Automatic 25x25 matrix ● Manual
MOD: ● Salesman ○ Start-Stop
ACO parameters: α 0.521 NCS 50; β 1.238 Ant number 50; ρ 0.693
Start-Stop points: Start 4 Stop 7

Best path: 4 3 2 1 0 5 6 7 8 9 10 13 14 15 16 12 11 17 18 19 20 21 22 23 24
Distance =4241

Figure 4. Simulation 3.

4. Conclusion

The performance of ACO is related to suitable parameters. Nevertheless, it is difficult to determine the parameters via trial and error method. In this study, we used a hybrid GA-ACO algorithm to find the best path for TSP. The parameters of ACO are determined via GA. GA-ACO can be used in problems where it is difficult to determine ACO parameters. Several experiments were done with developed software. Three of them are presented in the paper. It has seen that the parameters of ACO change for each experiment. The experimental studies showed the superiority of the method. The shortest route is found every time. This study will help the researchers who are making parameter optimization.

References

[1] S. Koziel and X.-S. Yang, Computational optimization, methods and algorithms, vol. 356. Springer, 2011.

[2] M. Mitchell, An Introduction to genetic algorithms. MIT Press, 1998.

[3] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Neural Networks, 1995. Proceedings., IEEE Int. Conf., vol. 4, pp. 1942–1948 vol.4, 1995.

[4] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, 2006.

[5] A. Mucherino, O. Seref, O. Seref, O. E. Kundakcioglu, and P. Pardalos, “Monkey search: a novel metaheuristic search for global optimization,” in AIP conference proceedings, 2007, vol. 953, no. 1, pp. 162–173.

[6] C. Yang, X. Tu, and J. Chen, “Algorithm of marriage in honey bees optimization based on the wolf pack search,” Proc. 2007 Int. Conf. Intell. Pervasive Comput. IPC 2007, pp. 462–467, 2007.

[7] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 2009, pp. 210–214.

[8] W. Pan, “Knowledge-Based Systems A new Fruit Fly Optimization Algorithm : Taking the financial distress model as an example,” Knowledge-Based Syst., vol. 26, pp. 69–74, 2012.

[9] A. Kaveh and N. Farhoudi, “A new optimization method: Dolphin echolocation,” Adv. Eng. Softw., vol. 59, pp. 53–70, 2013.

[10] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” Adv. Eng. Softw., vol. 95, pp. 51–67, 2016.

[11] X. Chen, Y. Kong, X. Fang, and Q. Wu, “A fast two-stage ACO algorithm for robotic path planning,” Neural Comput. Appl., vol. 22, no. 2, pp. 313–319, 2013.

[12] M. a. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, “Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation,” Appl. Soft Comput., vol. 9, no. 3, pp. 1102–1110, 2009.

[13] J. E. Aghazadeh Heris and M. A. Oskoei, “Modified genetic algorithm for solving n-queens problem,” 2014 Iran. Conf. Intell. Syst., pp. 1–5, 2014.

[14] A. C. Nearchou, “Path planning of a mobile robot using genetic heuristics,” Robotica, vol. 16, no. 5, pp. 575–588, 1998.

[15] A. C. Nearchou and N. A. Aspragathos, “A genetic path planning algorithm for redundant articulated robots,” Robotica, vol. 15, no. 2, p. S0263574797000234, Mar. 1997.

[16] J. Ni, K. Wang, H. Huang, L. Wu, and C. Luo, “Robot path planning based on an improved genetic algorithm with variable length chromosome,” 2016 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov., pp. 145–149, 2016.

[17] M. S. Saad, H. Jamaluddin, and I. Z. M. Darus, “Implementation of PID controller tuning using differential evolution and genetic algorithms,” Int. J. Innov. Comput. Inf. Control, vol. 8, no. 11, pp. 7761–7779, 2012.

[18] R. K. Tan and Ş. Bora, “Parameter Tuning Algorithms in Modeling And Simulation,” Int. J. Eng. Sci. Appl., vol. 1, no. 2, pp. 58–66, 2017.

[19] N. M. Razali and J. Geraghty, “Genetic Algorithm Performance with Different Selection Strategies in Solving TSP,” Int. Conf. Comput. Intell. Intell. Syst., 2011.