

# Combination of LabVIEW and Improved Ant Colony Algorithms for Optimization Path Design of Pneumatic Robot Manipulator

Wen-Jong Chen

Department of Industrial Education and Technology, National Changhua University of Education, Changhua, Taiwan  
(wjchen64@gmail.com)

**Abstract-** This article presents an improved ant colony optimization (IACO) algorithm to calculate the shortest path for pneumatic robot manipulator. MATLAB Script node in LabVIEW was used to determine the optimum trajectories and sequent nodes of moving for robot system. The LabVIEW graphical development software was used to construct the graphical user interference (GUI) of the robot manipulator, monitoring program, and the human machine. The results show that the shortest path improved by about 0.44%, 1.05% and 3.92%, and the running time by about 79.5%, 81.8% and 84.42% for 40 nodes, 90 nodes and 140 nodes, respectively. It can be clearly inferred from this conclusion that t with the increased number of trajectory nodes, the shortest path lengths and the running time of IACO algorithms are gradually less than those of TACO algorithms

**Keywords-** ant colony optimization; robot manipulator; path planning.

## I. INTRODUCTION

With the development of modern industry and the progress of scientific technological, the robot manipulator plays a very important role in an automatic manufacturing process. One aspect of interest in robot manipulator is the optimum motion path planning of the manufacturing processes to increase the productivity and the precision of robot systems. The precision of the process and quality may be improved at the programming level by defining the Cartesian path with more path points, employing a more sophisticated path generation method while the production rate, productivity, is improved by incorporating an efficient trajectory generation algorithm which results in the time optimal trajectories [1]. The paths of the robot manipulator are determined by the end-effector, and the positions of robot manipulator passing are equivalent to the locations of the cities ("nodes") in the travelling salesman problem (TSP). One of the most powerful techniques used in this field is the ant colony algorithm.

Ant colony algorithm is a new type of metaheuristic search to solve the combinatorial optimization problems. In several studies, the concept of ant colony algorithms was used to solve

the problems of the quadratic assignment; system identification, data mining, and IC design [2-8]. In addition, ant colony optimization (ACO) algorithm is one of the optimal algorithms for solving a routing scheme problem, and is also an effective method to find optimal solutions for difficult discrete optimization problems [10], especially in the path planning of robot motion.

Path planning is a key step in the control of mobile robot. And the quality of path influences in the efficiency of mobile robot. So designing an efficient path planning algorithm is essential. In past years, there are many algorithms for path planning have been researched. Tamilselvi *et al* [9] used the genetic algorithm approach to optimize path planning of a mobile robot in 2D workplace. They analyzed the path planning of collision-free and obstacle avoidance for a mobile robot in both static and dynamic environments. O.Hachour [10] presented an algorithm for path planning to a target for mobile robot in unknown environment to find the path in order to reach the target without collision. John *et al* [11] applied the ant colony optimization (ACO) to analyze the vehicle routing problems (VRP). They used the modified ACO to provide the shortest path for the multiple routes of the VRP. Michael *et al* [12] investigated the application of ACO to robot path planning in a dynamic environment. They presented two different pheromone re-initialization schemes to find the shortest and collision-free route. Yee Zi Cong and S. G. Ponnambalam [13] presented the ACO algorithm to test on a given set of maps for a mobile robot. They determined the shortest distance possible in travelling from the starting point to the destination.

In this paper, an improved ant colony optimization (IACO) algorithm with establishing a transient area of probabilities for all ants is proposed to solve the pneumatic robot manipulator path planning problem based on the global path planning approach. MATLAB Script node in LabVIEW was used as a computation/analysis tool to find the shortest path of the robot manipulator motion. A GUI function in LabVIEW connected experimental platform was built as a human machine interface (HMI). The experimental apparatus was a pneumatic robot manipulator that included 3 degrees of freedom (3-DOF) system to verifying the simulation results.

This article is organized as follows: Section 1 briefly introduces the background, motivation and the objective in this study. Section 2 describes the difference between traditional ant colony optimization (TACO) and the improved ant colony optimization algorithms (IACO), and the path principle of robot manipulator. Section 3 presents the experimental system. Section 4 presents the simulation examples and experimental results. Finally, Section 5 offers conclusions

## II. DESCRIPTION OF METHODS

### A. Traditional Ant Colony Optimization Algorithms

Traditional Ant Colony Optimization Algorithms (TACO), which are used to find the optimal path of probability-based algorithm, were first proposed by Italian scholar Dorigo et al [14]. It was inspired by the ants finding the shortest path from their nest to a food source, and vice versa. The ant colony algorithms imitate the techniques used by real ants to rapidly establish the shortest path, as shown in Figure 1. When foraging, ants leave a pheromone trail on their passed path. Because ants can smell pheromone, other randomly moving ants in the neighborhood can detect this marked pheromone trail. Numerous ants follow the pheromone-rich trail, and the probability of the trail being followed by other ants is further enhanced by increased trail deposition. This is an auto-catalytic process that favors the path along which more ants previously traversed. The ant system algorithms were based on the indirect communication capabilities of the ants. Artificial ants in the ACO algorithms are deputed to generate rules by using heuristic information or visibility, and by using the principle of indirect pheromone communication capabilities for iterative improvement of rules. [15-17] The flowchart of the TACO algorithms is shown in Figure 2, and the descriptions of various steps are presented as follows:

- *Step 1:* Randomly select the initial city (or “node”) for each of the ants, and set the initial state of the parameters.

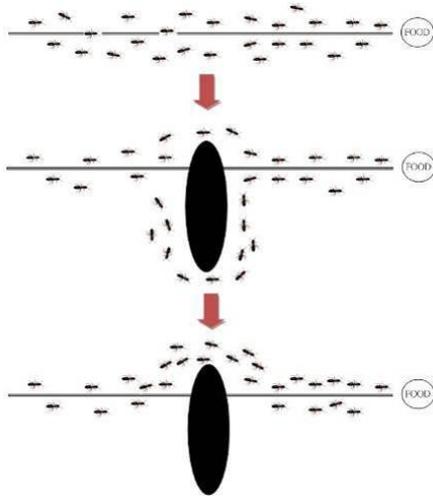


Figure 1. Real ants find a shortest path.

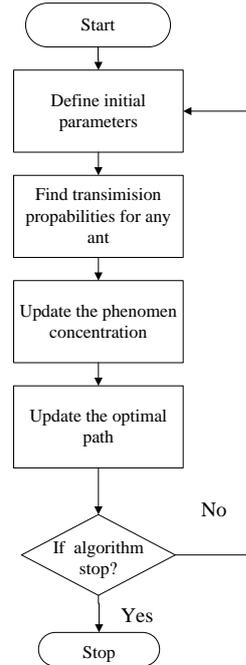


Figure 2. Flow chart of TACO.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{m \in J_k(i)} [\tau_{im}(t)]^\alpha \times [\eta_{im}]^\beta} ; & \text{if } j \in J_k(i), \\ 0 ; & \text{others,} \end{cases} \quad (1)$$

where  $P_{ij}^k(t)$  is the transition probability for the  $k^{\text{th}}$  ant at time  $t$ .  $i$  is the current city,  $j$  is the next city,  $\tau_{ij}(t)$  is the pheromone level between city  $i$  and city  $j$  at time  $t$ ,  $\eta_{ij}$  is the inverse of the distance between city  $i$  and city  $j$ ,  $J_k(i)$  is the set of cities that remain to be visited by the  $k^{\text{th}}$  ant positioned on city  $i$ , and  $\alpha$  and  $\beta$  are the parameters that determine the relative importance of pheromone level versus distance. The initial pheromone concentration ( $\tau_0$ ) between any two cities (from city  $i$  to city  $j$ ) is set as a small positive constant:  $\tau_{ij} = \tau_0 = (NL)^{-1}$ , where  $N$  is the number of the total segments between all of cities,  $L$  is the total length of all cities.

- *Step 2:* Calculate the transition probability  $P_{ij}^k(t)$  from city  $i$  to city  $j$  for the  $k^{\text{th}}$  ant.

All of the ants are put on randomly chosen nodes. Each ant starts from its node and obtains the probability to select the next node through the transition probability shown in formula (1). For each arrival node with every ant, formula (1) was repeated to find the passing path of every ant.

- *Step 3:* Update the pheromone concentration between cities.

When all ants complete a journey, the pheromone is updated in accordance with the behaviors of all ants. The

pheromone concentration on segment  $i$  and  $j$  left from the path passed by the ants is shown in formula (2),

$$\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \rho \sum_{k=1}^K \Delta\tau_{ij}^k \quad (2)$$

where  $0 < \rho < 1$  is a degenerating parameter of the pheromone,  $(1-\rho)$  is evaporation (i.e., loss) rate of pheromones, the symbol  $\leftarrow$  is used to show the next iteration, and  $\Delta\tau_{ij}^k$  is the left pheromone concentration (from  $t$  to  $t+1$ ) between cities  $i$  and  $j$  by the  $k^{\text{th}}$  ant. If  $Q$  is a parameter of the pheromone strength and  $L_{ij}^k$  represents the length between cities  $i$  and  $j$  visited by the  $k^{\text{th}}$  ant,  $\Delta\tau_{ij}^k = Q/L_{ij}^k$ . In general, the  $Q$  value is set to 100. Each ant repeats step 2 until all ants have toured all cities.

- **Step 4:** Update the optimal route.

The optimal route is updated and the shortest path is selected within all of the paths passed by the ants at time  $t$ . Subsequently, let the time  $t$  renew  $t+1$ . Because time  $t$  is the same as the iteration number in the iterative computation, time  $t$  equals the  $t^{\text{th}}$  iteration.

- **Step 5:** Test stop conditions.

From step (4), after all of ants have completed a trip and updated the pheromone concentration, the current iteration can proceed to the next iteration. The stop conditions are that the iteration is up to the maximal number set. If the iteration arrives at the maximal iterative set number, that is,  $t = t_{\text{max}}$ , the algorithm will stop. If not, the algorithm returns to step 2.

### B. Improved Ant Optimization Algorithms

In this section, we focus our attention on replacing the transition probability with a transient area. The improved ant colony optimization (IACO) algorithms are shown in Figure 3. The steps of IACO algorithms differ from those of TACO algorithms in managing transition probability. In IACO algorithms, the transient area is set to preserve all of the select probabilities of the  $k^{\text{th}}$  ant choosing the next city at time  $t$ ,  $trans_{ij}^k(t)$  represents the select probabilities between cities  $i$  and  $j$  for the  $k^{\text{th}}$  ant in transient area,  $\sum trans_{ij}^k$  represents the probability summation of each segment at the same path for the  $k^{\text{th}}$  ant at time  $t$ .  $Rand()$  is a random number [0,1], and the probability on segment  $i$  and  $j$  in transient area is as follows,

$$trans_{ij}^k(t) = [\tau_{ij}^k(t)]^\alpha \times [\Delta\tau_{ij}^k]^\beta \quad (3)$$

$$\Psi_{h+1}^k(t) = \Psi_h^k(t) + trans_{ih}^k(t) \quad (h = 1, 2, \dots, n) \quad (4)$$

$$r^k = Rand() \times \sum trans_{ij}^k(t) \quad (5)$$

where  $r$  is the next city. If  $i = j$ , the select probability is 0.  $h$  is the  $n$  sequent cities.  $\Psi_h^k(t)$  is initially defined as 0. If  $\Psi_{h+1}^k(t) > r^k$ , the  $k^{\text{th}}$  ant selects next city  $h$ . If not,  $h = h+1$  and go to Eq. (3)

$$\tau_{ij}(t_\rho) \leftarrow (1-\rho)\tau_{ij}(t) \quad (6)$$

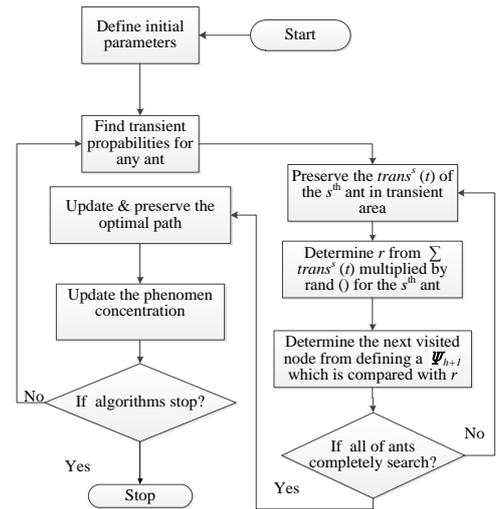


Figure 3. Flow chart of IACO.

$$\tau_{ij}(t+1) = \tau_{ij}(t_\rho) + 0.1 \quad (7)$$

Because the transient area preserves the probability of all ants choosing their paths, the next city is determined by the maximum probability in the transient area. If the pheromone concentration only evaporates but not increase, it will be less and less. Therefore, Eq.(7) needs to add to 0.1 to increase the pheromone concentration.

### III. TRAJECTORY DESCRIPTIONS OF ROBOT MANIPULATOR

In the algorithm processes, the end-effector of robot manipulator is similar to an ant. The ants passing path equals to the trajectories of the end-effector. The end-effector passing through the distance between two nodes can be depicted in Euclidean Space. The path of Euclidean distance between points  $A$  and  $B$  is the length of the line segment connecting them to  $\overline{AB}$ . In Cartesian coordinates, if  $A = (a_1, a_2, \dots, a_n)$  and  $B = (b_1, b_2, \dots, b_n)$  are two points in Euclidean  $n$ -space, the distance from  $A$  to  $B$  or from  $B$  to  $A$  is derived by formula (3).

$$\overline{AB} = \left[ \sum_{i=1}^n (a_i - b_i)^2 \right]^{1/2} \quad (3)$$

In three-dimensional Euclidean space, the distance formul between points  $A$  and  $B$ , and the  $a$  is as follows.

$$\overline{AB} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2} \quad (4)$$

### IV. SYSTEM CONSTRUCT

The signal sources to perform the motion of the robot manipulator are from the data acquisition (DAQ) unit connected to software systems of a PC. Subsequently, the experimental processes of the shortest path with the pneumatic robot manipulator are performed. In this study, the system can

fully display the path simulation and experimental verification.

### A. GUI Design

The National Instruments (NI) LabVIEW software was used to design the GUI between simulation and experimental apparatus. It provided the functions of data acquisition, data analysis, mathematical calculation, and interface controlling between software and hardware.

In the algorithm processes, the variables and formulae of the initial state, path nodes, transition probability, pheromone concentration, and path updating are presented in GUI. The input data, consisting of the coordinates of the robot manipulator passing nodes and all of the parameters, are shown in area (a) of Figure 4. Because the experimental apparatus is a RPP robot manipulator used a cylindrical coordinate system  $(r, \theta, z)$ , the coordinates  $r, \theta$  and  $z$  were converted into the Cartesian coordinate system  $(x, y, z)$ . The formula of coordinate conversion is shown in formula (5). The two cylindrical coordinates  $r$  and  $\theta$  can be converted to the two Cartesian coordinates  $x$  and  $y$  by using the trigonometric functions sine and cosine. Then the height  $z$  coordinate is the same in both systems.

$$\begin{aligned} x &= r \times \cos(\theta \times \pi / 180), \\ y &= r \times \sin(\theta \times \pi / 180), \\ z &= z, \end{aligned} \tag{5}$$

The data of the path trajectory in the control panel (areas (b), (c) and (d) in Figure 4) displayed the sequent path nodes and path length before and after optimization. The areas (d) and (e) in Figure 4 illustrate the path trajectories, path length, and subsequent path nodes through the calculation used in MATLAB Script. The areas (f) and (g) in Figure 4 demonstrate that the robot manipulator completes the motions and subsequently returns to the initial position through the transmitting signals of DAQ cards. In addition, area (h) on the control panel has six lamps, which represent, from the left of the control panel to the right, the states of the robot manipulator motion. When the robot manipulator moves, the lamps light according to the orders and directions of the pneumatic cylinder motion.

### B. Experimental Apparatus Installed

As shown in Figure 5, the main experimental platform consists of DAQ unit, amplified driving circuits, pneumatic circuits, and robot manipulator modules. When completing the calculations, DAQ unit transforms the position signals into the

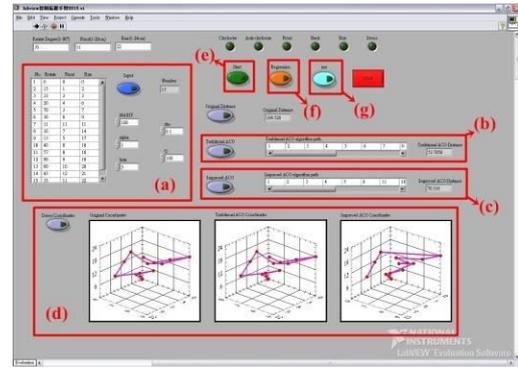


Figure 4. Path length and trajectories analysis on LabVIEW interface.

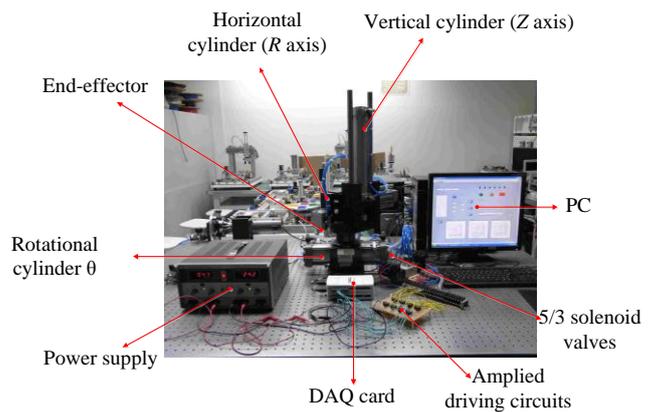


Figure 5. Hardware setup of experimental platform.

digital signal and subsequently controls the pneumatic circuits through amplifying the current apparatus. The controlling and calculating center of all system are operated on a Personal Computer with Pentium IV-3.2 GHz CPU and 512 MB SDRAM. Figure 6 illustrates the structure of the whole robot system.

1) *Pneumatic Robot manipulator*: The experimental system included RPP pneumatic robot manipulator and end-effector. The RPP pneumatic robot manipulator is suitable for the experimental platform of teaching and learning processes because of its rapid motion, simplified structure, and it is easy to assemble and disassemble. In industry applications, this type of robot manipulator is widely used in Warehouse Logistics Systems to locate and store materials. In Figure 7, the whole robot manipulator includes the vertical cylinder, horizontal cylinder, rotational cylinder, and the end-effector. The vertical and horizontal pneumatic cylinder can control the linear motion along the Z-axis and X-axis, respectively. The rotational pneumatic cylinder controls the rotational angle of Z-axis and the end-effector controls the action of gripping materials.

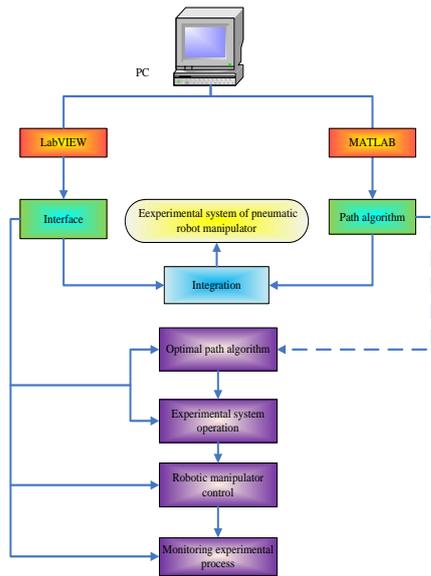


Figure 6. Schematic diagram of the whole system structure.

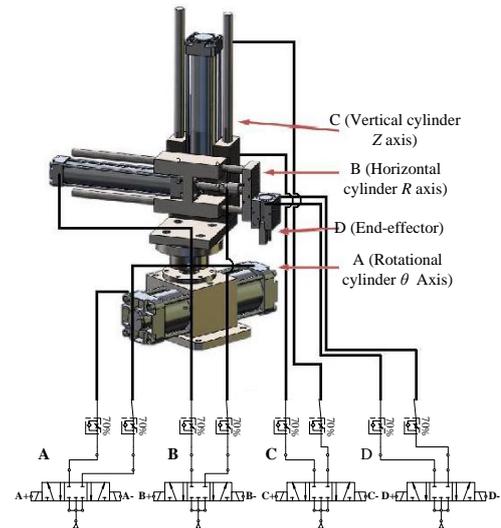


Figure 7. Pneumatic Robot manipulator.

2) *Applied Driving Circuits*: This control system and components include four series of pneumatic circuits, as shown in Figure 7. Each of the circuit systems consists of an actuator, a AC 110V 5/3 solenoid valve, which controls the movement of a plunger, and a one-way throttle valve, which is a speed control device. The one-way throttle valve controls the flow-rate within the cylinder during movement of the actuator. That is, it can influence the moving speed of the actuator. Each of the 5/3 solenoid valves can control 3 positions of one actuator, including forward, backward, and mid-position. It enables more flexible and complete path planning for the robot manipulator. The air enters the solenoid valve when the START button is pressed, and the solenoid coils in the valve are excited when currents pass through. The gas flows to the cylinders of the actuators, and subsequently, the robot manipulator was driven to complete the entire motion.

3) *Data Acquisition*: The DAQ unit that was used for creating a signal transmission interface is a NI USB-6212 card, which includes 4 digital I/O channels, 16-bit resolution, and 250 kS/s sampling rate to transmit the signals to control the solenoid valves. The signals through the DAQ unit must be amplified by driving the circuit modules to drive the pneumatic circuit in robot manipulator. Because the robot manipulator has 4-pairs of motion models (3-DOF and end-effector motions), two DAQ cards are required in this system. Each of the digital I/O per DAQ card controls one pair of motion models for the robot manipulator.

4) *Amplified Driving Circuit*: Because the currents in the DAQ unit are relatively low and unstable, a group of amplified circuit modules, named Darlington Circuit (TIP 120 NPN Darlington Transistor), were applied to avoid abnormal motion of the robot manipulator. Figure 5 illustrates the amplified current derived from Darlington Circuit. The objective of the amplified output currents was to drive AC 110 V solenoid

valves to control the directions of the actuator in the robot manipulator.

## V. RESULTS AND ANALYSIS

In this study, the degenerating parameter of the pheromone  $\rho$  was set to 0.1 and the parameters  $\alpha$  and  $\beta$  were set to 1 and 5 in TACO and IACO algorithms, respectively. To demonstrate the comparison of the IACO and TACO, we used 40, 90 and 140 nodes to verify the difference between the TACO and IACO algorithms. The simulation and experimental results of the robot manipulator are depicted in Figures 8-13. Figures 8-10 show the convergence diagrams of 40, 60 and 90 nodes of TACO and IACO algorithms. When the trajectories of TACO and IACO for robot manipulator are considered, it is seen that the shortest path lengths of IACO are better than those of TACO. In addition, the running time of the IACO algorithms is far less than that of TACO algorithms; that is, the IACO algorithms utilized the transient area of transition probabilities for each ant to increase the chance of obtaining the global optimal solutions. It allowed the search to avoid premature convergence to decrease the risk of falling into the optimal solution of the local zone and to search for a global optimal solution in the search space. As shown in Figure 11-13, with the increased number of trajectory nodes, the shortest path lengths and the running time of IACO algorithms are gradually less than those of TACO algorithms, and the path lengths and running time of IACO algorithms are less than those of the TACO algorithms at the same iterative number. As shown in Table 1, the shortest path lengths improved by about 0.44%, 1.05% and 3.92%, and the total running time by about 79.5%, 81.8% and 84.42% for 40, 90 and 140 nodes, respectively. Obviously, it can be observed that the IACO algorithm through the simulation and experimental verification greatly improved its global search capability without losing its fast convergence property.

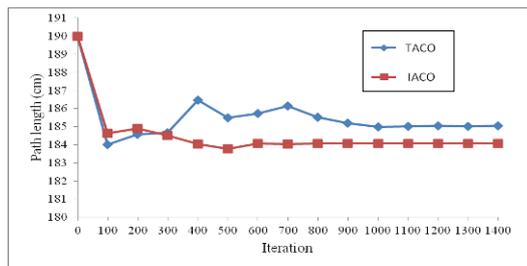


Figure 8. Comparison of TACO and IACO for 40 nodes.

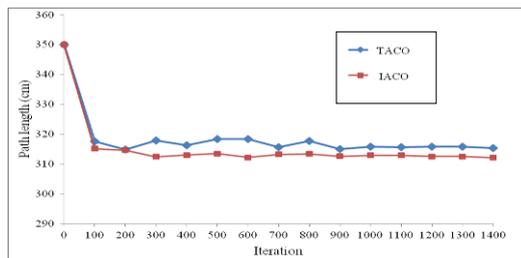


Figure 9. Comparison of TACO and IACO for 90 nodes.

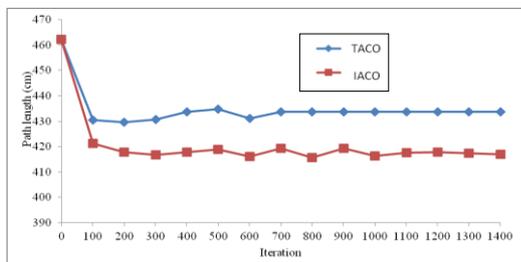


Figure 10. Comparison of TACO and IACO for 140 nodes.

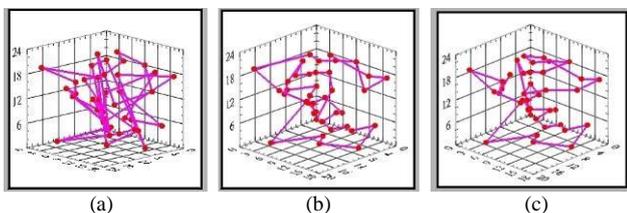


Figure 11. Trajectories of 40 nodes (a) Original path (b) TACO path (c) IACO path.

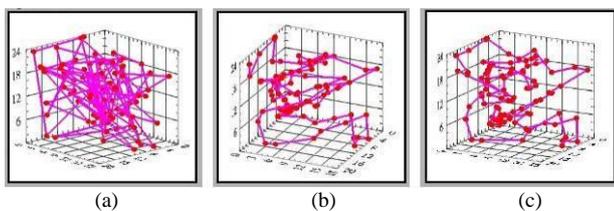


Figure 12. Trajectories of 90 nodes (a) Original path (b) TACO path (c) IACO path.

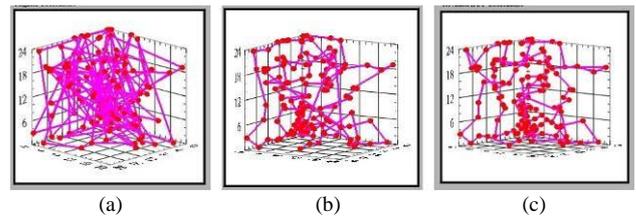


Figure 13. Trajectories of 140 nodes (a) Original path (b) TACO path (c) IACO path.

Table I. EXPERIMENTAL RESULTS OF TACO AND IACO.

Nodes	Items	Original path	TACO	IACO
40	Path length (cm)	191.1	184.9	184.1
	Time (sec)		101.9	25.0
90	Path length (cm)	351.3	315.3	312.1
	Time (sec)		2119.4	391.2
140	Path length (cm)	462.1	433.8	417.0
	Time (sec)		9382.3	1468.5

## VI. CONCLUSION

In this paper, we have proposed an improved ant colony optimization algorithm to promote the performance of robot system, which is combined the LabVIEW and pneumatic robot manipulator in global search. We can get better search results using this improved algorithm. In this improved algorithm, the original paths are distributed randomly within the moving range of robot system, a global searching is assured around the global optimum and shorten the total running time. Therefore, the improved algorithm can be used to avoid wasting too much working time for the automatic production. This indicates that the improved algorithm in this study can also be employed to automatic process to promote the productivity, quality and competitiveness.

## REFERENCES

- [1] Z. S. Abo-Hammoura, N. M. Mirza, S. M. Mirza, M. Arif, "Cartesian Path Generation of Robot Manipulators Using Continuous Genetic Algorithms," *Robot Auton. Syst.*, Vol. 41, pp.179–223, 2002.
- [2] X. Zhang, L. Tang, "A New Hybrid Ant Colony Optimization Algorithm for The Vehicle Routing Problem," *Pattern Recogn. Lett.*, Vol. 30, pp. 848–855, 2009.
- [3] K. Gheysari, A. Khoei, B. Mashoufi, "High speed ant colony optimization CMOS chip," *Expert Syst. Appl.*, Vol.38, pp. 3632–3639, 2011.
- [4] S. C. Chu, J. F. Roddick, J. S. Pan, "Ant Colony System with Communication Strategies," *Inform. Sciences*, Vol. 167, pp. 63–76, 2004.
- [5] J. Heinonen, F. Pettersson, "Hybrid Ant Colony Optimization And Visibility Studies Applied to A Job-Shop Scheduling Problem," *Appl. Math. Comput.*, Vol. 187, pp. 989–998, 2007.
- [6] S. S. Weng, Y. H. Liu, "Mining Time Series Data for Segmentation By Using Ant Colony Optimization," *European Journal of Operational Research*, Vol. 173, pp. 921–937, 2006.
- [7] B. Biswal, P.K. Dash, S. Mishra, "A Hybrid Ant Colony Optimization Technique for Power Signal Pattern Classification," *Expert Syst. Appl.*, Vol. 38, pp. 6368–6375, 2011.
- [8] F. A. Chegury Viana, G. I. Kotinda, D. A. Rade, V. Steffen, "Tuning Dynamic Vibration Absorbers by Using Ant Colony Optimization," *Comput. Struct.*, Vol. 86, pp.1539–1549, 2008.

- [9] Tamilselvi, M. shalinie, Hariharasudan, "Optimal Path Selection for Mobile Robot Navigation," *IJCSI International Journal of Computer Science Issues*, Vol. 8(4), pp. 433-440, 2011.
- [10] O.Hachour, "Path planning of Autonomous Mobile robot," *International Journal Of Systems Applications, Engineering & Development*, Vol. 2(4), pp. 178-190, 2008.
- [11] John E. Bell, P. R. McMullen, "Ant Colony Optimization Techniques for The Vehicle Routing Problem," *Adv. Eng. Inform.*, Vol. 18, pp. 41-48, 2004.
- [12] Michael Brandl, Michael MasudaI, Nicole Wehner, Xiao-Hua Yu, "Ant Colony Optimization Algorithm for Robot Path Planning," *International Conference on Computer Design And Appliations* , Vol.3, pp.436-440, 2010.
- [13] Y. Z.i Cong, S. G Ponnambalam, "Mobile Robot Path Planning using Ant Colony Optimization. 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center. Singapore, July 14-17,pp. 851-856, 2009.
- [14] M. Dorigo, "Optimization, Learning And Natural Algorithms," PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.
- [15] Zar Chi Su Su Hlaing, May Aye Khine. An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem. 2011 International Conference on Information Communication and Management, Vol.16, pp. 54-59, 2011.
- [16] Z. Q. CAI, H. HUANG, Y. QIN, X. H. MA, "Ant Colony Optimization Based on Adaptive Volatility Rate of Pheromone Trail," *Int. J. Communications, Network and System Sciences*, Vol. 2, pp.792-796 , 2009
- [17] A. H. Borzabadi, H. H. Mehne, "Ant Colony Optimization for Optimal Control Problems," *Journal of Information and Computing Science* Vol. 4(4), pp. 259-264, 2009.
- [18] S. Kaur, P. Agarwal, R.S. Rana, "Ant Colony Optimization :A Technique used for Image Processing," *International Journal of Computer Science and Technology*, Vol. 2(2), pp.173-175, 2011.