THE USE OF ANT COLONY OPTIMIZATION IN COMMERCIAL TRAINING ALLOTMENT

UMAKANTA DASH Dr. N.Mohana Suganthi Department of Computer Science & Engineering
Raajdhanni Engineering College

Abstract—
The most common way of relegating a meeting college's manager to visit a gathering of modern preparation pragmatic understudies in the college is at present being done physically. To perform such errand, two requirements should be satisfied whenever: (1) Practical understudy must be managed by a college boss from a similar division; (2) the area of the spots to be visited by the meeting college's manager should be pretty much as close as conceivable to streamline the voyaging cost, time and financial plan. Utilizing the manual methodology, the interaction can be exceptionally drawn-out and tedious particularly when it included huge number of reasonable understudies and teachers. Besides, the enhanced outcome is rarely attainable as not all useful understudy teacher blends are inspected. Via mechanizing the interaction, the dreary and tedious cycle can be tried not to as well as lay out enhanced mixes in view of the given imperatives. This paper talks about on how the task interaction is computerized utilizing Ant Colony Optimization (ACO). The outcomes are then contrasted with Dijkstra's Algorithm with assess the capacity of ACO calculations. The calculation plan, execution, its future course and upgrades are talked about aswell.

IndexTerms—
AntColonyOptimization;AllocationProblem; Industrial Training; Path FindingProblem.

I. INTRODUCTION

One of the graduation requirements for Bachelor practical studentsintheuniversityistoundergoforindustrialtraining. Practical students who registered for industrial training will be attached to a company to gain experiences and get expose totheworkenvironment.Thecompanyororganizationwhich hosts the practical student will assign appropriate projects or taskstothepracticalstudentstoduringthetrainingperiod.At theendofthetraining,practicalstudentsneedtopresenttheir progress for the whole training duration. The presentation will be done at the training location, with the present of a visiting lecturer from theuniversity.

The Industrial Training Committee (ITC) who is responsible to manage the process will assign lecturers as visitinglecturerfortheassessmentorpresentationsession.In order to optimize travelling cost and time, each lecturer will be assigned to a few practical students in the same vicinity. The process of assigning lecturer to practical students could be tedious and time taxing to the committee. This is due to various locations of practical students throughout the nation and in some cases; the location is quiet far from one another. Sincetheprocessisdonemanually,theallocationofpractical students-lecturers could be not optimized andinefficient.

In this paper, we propose Ant Colony Optimization(ACO) to be used for assigning lecturers to visit industrial training practicalstudents.ACOischosenbasedonitsabilitiestofind approximate solutions for optimization problems, using software agents known as artificial ants.

The objective of this study is divided into two. First, we apply ACO to allocate a suitable university's supervisor to a few practical students based on few constraints. The constraints that need to be fulfilled are:

1. A university's supervisor (visiting university lecturer) must supervise a certain number of practical students from the same academicdepartment.
2. The location of the places to be visited by the university university's supervisor must as near as possible in order to help the university's supervisor plantheirjourney,timeandoptimizedtheuniversity's budget.

For the second objective, we compared the ability of ACO in solving the problem with a known shortest pathalgorithm, Dijkstra'sAlgorithm.Timetakentofindtheshortestpathand the distance between places are recorded for both algorithm to evaluate the performance ofACO.

Inthenextsection,wepresenttherelatedworks.InSection 3, the implementation of ACO in the industrial training allocation system is discussed. In Section 4, we discuss the results of implementing the ACO. And lastly in the last section we present our conclusion and propose the future works.

II. LITERATUREREVIEW

A. *Ant Colony Optimization(ACO)*

ACOwasintroducedbyDorigo[1]intheapplication ofthe classical Travelling Salesman problem. The algorithm inspired by the foraging behavior of real ants in search of foods. Among the capabilities of real antsare:

1. Find the shortest path from a food source to the nest without using visual cues[2].
2. Adapt to change in theenvironments.

The main characteristic of this model [3] are positive feedback that accounts for rapid discovery of the new solutions, distributed computation to avoids for premature convergence and the use of constructive greedy heuristicthat helps find acceptable solutions in the early stages of the searchprocess.

B. *Behavior of ACOAlgorithm*

An ant travels from the colony in search of food source, whenanantfindsafoodsourceitleavesatrailofpheromones along the way from the food source to the colony. Other ants wouldfollowthistrailpheromonestoexploitthefoodsource. Thestrongertheconcentrationofpheromonesinthetrail,the more ants will followit.

This process can be broken down into the following steps:

1. The ant leaves the colony searching forfood
2. Since it is the first ant, it searches the surrounding environment randomly
3. Whentheantfindsafoodsource,itwillgobacktothe colonyleavingatrailofpheromonesonthewayback.
4. Otherantssearchforfoodsourcesinaslightlyrandom manner, they follow the trail with the strongest pheromone concentration. Yet, some ants might follow other paths searching for other food sources that might be better.
5. The pheromones on the trails vaporizes in time, this indicates that trails that have a strong pheromone concentration are the most used trails. If the food sourceisdepletedornolongerbeneficial,lessantswill travel to it and eventually the pheromone trail would disappear.

At first, an ant would leave the colony N searching randomlyforfoodsourceF.Itrandomlychoosesapathtogo through to reach the food source, on its way back to N, and the ant would leave a trail of pheromones. The next ants will take the path with the pheromone trail or choose to explore a different path to the food source. After a while, the pheromone trail on the path with less ants will start to evaporate, which reduces the path's attractiveness to other ants. The longer the path between the food source and the colony, the more time the pheromone has to evaporate. In situation where more ants are traveling the short path to the food source, the longer path is slowly abandoned. Figure 1 illustrate the behavior of ACO algorithm.

This natural behavior of ants has inspired the main characteristics of this organizational model, which are distributedcomputationforavoidingprematureconvergence, sharing positive feedback that helps in quick discovery of new solutions, and finding acceptable solutions early in the search process using greedy heuristics[4].

Figure 1: Behavior of ACO Algorithm

C. Application ofACO

Since ACO was first proposed, number of research and application adopting this technique has been increased.

Among the popular applied domain area are routing, assignment and scheduling, For routing, ACO has been applied in various problem such as travelling salesman[5-7], vehicle route planning [8-12], emergency escape and evacuation planning [13-16], also transportation problem [17,18].

III. IMPLEMENTATION OF ACO FOR INDUSTRIALTRAINING ALLOCATION

1. *The Input*

Details of university's supervisors and practical students will be needed to be the input to the algorithm parts such as name, ID number; faculty and home address and coordinate oftheaddress(Figure2).Practicalstudent'sdetailsar ename, ID number, faculty, company address and coordinate(Figure 3). Coordinate which is the latitude and longitude is used instead of the real because it can provide accurate result and show the exact starting point and end point for the distance calculation.

Figure 2: Interface of inserting new lecturer details

Figure 3: Interface of inserting new practical students details

2. *TheImplementation*

The implementation is done by using Graph Hopper application. This application allows the developer to convert theA*algorithm,abuilt-inalgorithm,toACOandDijkstra's Algorithm. Dijkstra's Algorithm will be used as comparison to evaluate the ability forACO.

Several paths will be creating by the ants from starting point to the destination in order to identify the shortest path that can be used to reach the food destination. In this study, startingpointreferstothelocationofuniversity'ssupervisor. The location can be their home's address during the initial allocation or location of the first practical student that have been assigned previously to the university's supervisor. Meanwhile, destination refers to the company's address where the practical students have beenplaced.

Once the shortest path has been found, all the ants will use thesamepathtomovefromstartingpointtoendpoint.Same as ant's movement, several paths will be finding by the ant colony algorithm from university's supervisors location to practical students location. Then the algorithm will identify the shortest path among the founded path in order to allocate the practical students to thelecturers.

All the searching process of the random paths and allocation of the shortest path from starting point to ending point will be process internally by the algorithm. So as a result, the founded shortest path will be display as an output due to time efficiency (Figure 4).

Figure 4: Display output

To ensure the accuracy of the ACO results, Dijkstra's Algorithm is applied to the same test cases. Dijkstra's was chosen based on its powerful ability in calculating shortest distance path for various research previously. The flowchart of the implementation as illustrated in Figure5.

Figure 5: Flowchart of theimplementation

IV. RESULTS

Ten test cases have been conducted. As mentioned earlier, one of the problems aimed to be solved is the extensive time required to complete the allocation process manually. Thus, time extraction for both algorithm to perform the allocation process is recorded.

The first objective for this study is the algorithm must be able to allocate a university supervisor's must be in the same academicdepartmentwiththepracticalstudents.Asexample, if the university supervisor's is from Software Engineering department,theallocatedpracticalstudentmustalsoa student from Software Engineeringdepartment.

At the same time, the algorithm has to ensure that the location of the places to be visited by the university university's supervisor must as near as possible in order to help the university's supervisor plan their journey, time and optimized the university's budget. Based on both objectives, we create several test cases to measure the achievement.

Test case 1 contains data with two university'ssupervisors (L1 and L2) and two practical students (S1 and S2) from the same department. Result from the test case shown that the allocation was given according to the department and asnear as possible from the supervisor's location (Table1).

Table 1
Results for
Test Case1

Lecturer	Results from ACO		Results from Dijkstra	
	Student	Distance (km)	Student	Distance (km)
L1	S2	8.783	S2	8.783
L2	S1	19.435	S1	18.656

Test case 8 contains data with four university supervisors (L1, L2, L3 and L4) and five practical students (S1, S2, S3, S4 and S5), also from the same department. The allocation are fair to all university supervisor except for L1 since the number of student are greater than number of university supervisorby1(Table2).TheallocationofS5istoL1dueto the nearer location compared with other university's supervisor.

Table 2
Result for Test Case 8

Lecturer	Results from ACO		Results from Dijkstra	
	Student	Distance (km)	Student	Distance (km)
L1	S2	8.783	S2	8.783
L1	S5	37.674	S5	30.96
L2	S3	11.928	S3	11.573
L3	S1	13.448	S1	11.139
L4	S4	4.95	S4	1.592

Additionaltestcase,whichistestcase10wasconductedto test the algorithm's ability in allocating data of university supervisor and practical student from different academic department (Table3).

Table 3
Data for different academic department

University Supervisor		Practical student	
ID	Department	ID	Department
L1	SE	S1	SE
L2	IS	S2	SE
L3	SE	S3	IS
L4	SN	S4	SN
		S5	SE
		S6	SN

Theresultsshowedthattheallocationweredoneaccording tothedepartment(Table4).AlthoughlocationS6isnearerto L1, the allocation is given to L4 due to same academic department.

Table 4
Result for Test case 10

Lecturer	Results from ACO		Results from Dijkstra	
	Student	Distance (km)	Student	Distance (km)
L1	S2	8.783	S2	8.783
L1	S5	37.674	S5	30.96
L2	S3	11.928	S3	11.573
L3	S1	13.448	S1	11.139
L4	S4	4.95	S4	1.592
L4	S6	151.496	S6	111.253

ResultsfromthetestcasesshownthatACOabletoperform the allocation process and produce the same allocation result as Dijkstra's algorithm. However there are variance in distance calculated by both algorithm. ACO takes longer times in producing the results), whereas Dijkstra's algorithm takesshortertimetoperformtheallocationprocess(Table5).

Table 5
Time extraction

Test Number	Total Lecturer	Total practical student	Total extract time by ACO	Total extract time by Dijkstra
1	2	2	0.0131	0.0064
2	2	2	0.0010	0.0007
3	2	2	0.0009	0.0009
4	2	2	0.0016	0.0004
5	3	3	0.0013	0.0011
6	4	4	0.0022	0.0042
7	4	4	0.0085	0.0053
8	4	5	0.0181	0.0119
9	5	5	0.0267	0.0159
10	5	6	0.0204	0.0090

However, in some condition, both of the algorithm complete the allocation process in similar time. In this study, the results shown that ACO is not good enough to perform the allocation process compared to Dijkstra's algorithm because the duration to perform the allocation process is slower when involves large number of data and. Furthermore, ACO searches for longer route compare to Dijkstra's algorithm.

V. CONCLUSIONS AND PROPOSED FUTUREWORKS

In this study, we have considered industrial training allocation problem to be solved using ACO. The objectives of this study are:

1. To allocate a suitable university's supervisor to a few practical students based on the definedconstraints
2. To evaluate the ability of ACO in performing the allocation process bycomparing the performance with Dijkstra'salgorithm.

We implement the algorithm by using the GrassHopper Application. Results from the test cases are then analyzed to evaluate the objectives.

Based on the results, we can concluded that:

- ACO is not suitable to be used in this study because it takes long time to perform the allocationprocess
- Dijkstra's algorithm provide optimum result in a shorter time
- The process involving more data requires more time forACO

In future, we propose to apply other heuristic algorithm such as Bee Colony Algorithm. Results from the study will be compared to investigate the efficiency of the algorithms. Wealsoplantoconductthetestcasewithlargerscaleof data in order to evaluate the efficient performance for both algorithm.

REFERENCES

[1] Colorni A., Dorigo M., and Maniezzo V. 1992. An Investigation of some Properties of an``Ant Algorithm'. *Ppsn*.(92):2–7.

[2] Guo Q., Ni S., Li C., and Chen D. 2009. An Improved Ant Colony AlgorithmforPlacing-InandTaking-OutofWagonsinBranch-Shaped Sidings. in *ICCTP*.*(*1996):1–6.

[3] Tsai C., Wu H., and Tsai C. 2002. A new data clustering approach for data mining in large databases. *Proc. Int. Symp. Parallel Archit. Algorithms Networks. I-SPAN'02*.315–320.

[4] Agrawal K. and Bagoria R. 2014. a Nt C Olony O Ptimization : E Fficient W Ay T O F Ind. *Int. J. Adv.Technol. Eng. Res.* 4(3):18–21.

[5] DorigoM.andGambardellaL.M.1997.Antcolo niesforthetravelling salesman problem. *Biosystems*. 43(2):73–81.

[6] Bianchi L., Gambardella L. M., and Dorigo M. An Ant Colony Optimization Approach to the Probabilistic Traveling Salesman Problem. in *Parallel Problem Solving from Nature - PPSN VII*.2439.

[7] Dorigo M., Maniezzo V., and Colorni A. 1996. Ant system: optimizationbyacolonyofcooperatingagents.*I EEETrans.Syst.Man Cybern. Part B.* 26(1)29–41.

[8] Tsou M. C. and Hsueh C. K. 2010. The study of ship collision avoidancerouteplanningbyantcolonyalgorithm. *J.Mar.Sci.Technol.* 18(5):746–756.

[9] Ding Q., Hu X., Sun L., and Wang Y. 2012. An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing*.98:101–107.

[10] Bell J. E. and McMullen P. R. 2004. Ant colony optimization techniques for the vehicle routing problem. *Adv. Eng. Informatics*. 18(1):41–48.

[11] YuB.,YangZ.-Z.,andXieJ.-X.2011.Aparallelimprovedantcolony optimization for multi-depot vehicle routing problem," *J. Oper. Res. Soc.* 62(1):183–188.

[12] Kammoun H. M., Kalle I.l, Alimi A. M., and Casillas J. 2010. An adaptive vehicle guidance system instigated from ant colony behavior. *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern*.2948–2955.

[13] GoodwinM.,GranmoO.,RadiantiJ.,SarsharP.,andGlimsdalS.2013. Ant Colony Optimisation for Planning Safe Escape Routes. in *Recent Trends in Applied Artificial Intelligence*.53–62.

[14] Goodwin M., Granmo O.-C., and Radianti J. 2015. Escape planning in realistic fire scenarios with Ant Colony Optimisation, *Appl. Intell.* 42(1):24–35.

[15] BaharmandH.andComesT.2015.AFrameworkforShelterLocation DecisionsbyAntColonyOptimization.in*ProceedingsoftheISCRAM 2015Conference*.

[16] Yan G. and Feng D. 2013. Escape-Route Planning of Underground Coal Mine Based on Improved Ant Algorithm. *Math. Probl. Eng.* 1– 14.

[17] Musa R., Arnaout J.-P., and Jung H. 2010. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Comput. Ind. Eng.* 59(1):85–92.

[18] ZhanZ.H.,ZhangJ.,LiY.,LiuO.,KwokS.K.,IpW.H.,andKaynak O. 2010. An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem, *IEEE Trans. Intell. Transp. Syst.* 11(2): 399–412.