



Visión Electrónica

Más que un estado sólido

<http://revistas.udistrital.edu.co/ojs/index.php/visele/index>



A RESEARCH VISION

Ant colony optimization algorithm for facility layout problem

Algoritmo de optimización de colonias de hormigas para el problema de distribución en planta

Jhineth Paola Obando Solano.¹, Jhonny Andrey Zamora Moreno.², Frank Nixon Giraldo Ramos.³

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:

Enviado: 06/05/2016

Recibido: 05/06/2016

Aceptado: 13/07/2016

Keywords:

metaheuristics

Combinatorial optimization

Ants Colonies Optimization

Facility Layout Problem.



Palabras clave:

Metaheurísticas

Optimización combinatorial

Optimización con colonias de hormigas

Problema de distribución en planta.

ABSTRACT

This paper is the result of a research on the applications of bio-inspired algorithms in the field of production engineering in the District University Francisco José de Caldas, covering the topics of operations research, industrial layout distribution in manufacturing plant among others. It is intended to seek optimization problems of these fields, using artificial intelligence of swarms from the implementation of an ant colony optimization algorithm (Ant Colony Optimization - ACO) as metaheuristic planning tool and optimization of layout problem. It plans, with the goal of finding the best spatial allocation of work stations or cells. Theoretical concepts explored and results are presented. First, a state of the art review on the subject made, then the possible solution algorithms were evaluated to identify the objective function to optimize, to finally apply the ACO algorithm, and evaluate the results of performance against the Initial configuration as the plant.

RESUMEN

Este artículo es resultado de un trabajo de investigación sobre las aplicaciones de algoritmos bioinspirados e inteligentes en el ámbito de ingeniería en producción, en la Universidad Distrital Francisco José de Caldas, abarcando las temáticas de investigación de operaciones, distribución en planta industrial de manufactura entre otros. Se pretende buscar la optimización a problemas propios de esos campos, aplicando inteligencia artificial de enjambres, a partir de la implementación de un algoritmo de Optimización de Colonias de Hormigas (Ant Colony Optimization - ACO) como herramienta metaheurística de planificación y optimización del problema de distribución en planta, con el objetivo de buscar la mejor asignación espacial de estaciones o celdas de trabajo. Se presentan los conceptos teóricos explorados y los resultados obtenidos. En primer lugar se efectuó la revisión de estado del arte sobre la temática, luego se evaluaron los posibles algoritmos de solución, para identificar la función objetivo a optimizar, para finalmente aplicar el algoritmo ACO, y evaluar los resultados de desempeño del mismo frente a la configuración inicial que tenía la planta.

¹Banco de Occidente. Bogotá, Colombia. E-mail: paitoobando@gmail.com

²IDARTES. Bogotá, Colombia. E-mail: jazamora85@hotmail.com.

³Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. E-mail: fngiraldor@udistrital.edu.co.

1. Introduction

The plant layout problem, is the design of the location or most optimal and flexible configuration of machines, equipment, resources, and physical space available, to facilitate the movement and handling of material and enable the plant to have a performance and optimized flow production, all under a minimum production cost and total time [1].

Traditionally, the study of such problems of distribution and spatial flow and optimization, has been addressed by dynamic programming techniques and combinatorial optimization, [2, 3] and in operations research [4, 5] it is often called “Facility Layout Problem - FLP”, [6–8].

The Ant Colony Optimization Algorithms, are adaptive and use metaheuristic techniques (general purpose heuristic algorithms that can be applied to different problems with minor modifications), inspired in the emulation of the behavior of wild ant colonies, and in organization for finding food resources with the goal of survival of individuals [9]. The behavior in which the ACO algorithm is based is called foraging or exploration. When searching for food, ants initially explore around its nest randomly. Ants communicate to others about the location of a food source, marking the path as they move with a trace of a substance called pheromone, which evaporates slowly. When they return to the nest, they leave a greater or lesser concentration of pheromone depending on the quantity and quality of food found around the explored place. Therefore, other ants follow the better path to the reliable sources of food, because the paths with higher concentration of pheromones have a greater probability of finding both greater quantity and better quality of food, while the other paths that are too far or with low probability of having a good food source, will have little traffic and therefore will diminish the concentration and intensity of pheromone trail by evaporation. They can apply, in convergence solution problems [10]. From the pheromone trails we can get a progressive and distributed optimization, from the contributions of each ant in the colony in search for the best solution.

2. Methods

This paper focuses on the application of ant colony optimization (ACO), as a tool for solving combinatorial optimization problems, such as the facility layout problem in industrial manufacturing plant. With this goal, there are many ant colony algorithms, like the Coloni, Dorigo and Maniezzo [7, 11, 12]. The idea is that

if the best solution is not guaranteed, then the optimal solution is reached. In this case the optimization can be performed considering that the following requirements include:

- a. Minimizing material handling and its total distance between machines and layout stations.
- b. Minimization of material flow.
- c. Minimizing the total distance traveled by the material.

2.1. Parameters, ants and nodes initialization

From the map of the layout that contains the (x, y) coordinates corresponding to the location of each of the nodes or workstations that are distributed on the plant, paths between nodes are defined, which form a set of initial solutions that ants must optimize in each cycle. Ants are initialized randomly in nodes defined by coordinates (x, y) , from each node progressively, construct each paths, until reach the solution optimized.

Additionally, each path on the problem is given a small amount of pheromone, $\tau_{ij} = \tau_0$, at each iteration, after the pheromone update is performed, a check is done to find out if its value is bounded in the interval (τ_{min}, τ_{max}) , the pheromone evaporation rate ρ .

2.2. Objective function for evaluating the performance of tours

All paths are initially proposed, they are evaluated by an objective function to determine its quality and rate their degree of specific setting (fitness). Which consists in this case, the evaluation of the sum of each of the distances between nodes in the path, to determine the best paths sequences adapted according to their score. Then by the foraging, ants verify the quality of each path optimized in each cycle. With this purpose, the Euclidean distance objective function is proposed, as the sum of the distance between the coordinate points of the current node (x_i, y_i) and the next node (x_{i+1}, y_{i+1}) , to complete the all nodes or workstations forming the layout to obtain the total length of each of the raised paths, (1):

$$ij = \sum_{i=1}^n \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Euclidian distance objective function

2.3. Evaluation of the cost of each solution

Heuristic information, called visibility is a function of the cost assignment. This cost can be represented by the flow and distances matrix. The heuristic information may be determined as, (2):

$$\eta_{ij} = \frac{1}{1 + d_{ij}} \quad (2)$$

Heuristic information function

Where d_{ij} corresponds to the distance between nodes or cost objective function, in this case it corresponds to the Euclidean distance between points [13].

2.4. Construction solution from pheromone trail

In this step we can define the trail of pheromones, from an equation that models the probability of transition between nodes of each solution path that are evaluated, thereby

marking the best paths. Taking τ_{ij} as the concentration or total amount of pheromone

deposited on each of the paths ij

at time t , and taking η_{ij} as a function that assigns a value heuristic (heuristic information) as the best possible solution for each ij path at time t according to the measure of the objective function. The probability that each ant is transferred between nodes is transitioning from a node i to node j , for a period of time t will be, (3)

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{i \in q} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, & \text{foraging } q \\ 0, & \text{other case} \end{cases} \quad (3)$$

Nodes Probability of transition per cycle

Where α is a parameter that determines the relative importance of the information of the pheromone, that is his visibility when prioritizing some paths against each other, and β is a parameter that determines the relative weight corresponding to the heuristic information trail, q is the set of all solution paths between nodes.

2.5. Updated trail pheromones on each path

By looking at this step, we can increase the level of pheromone in each of the paths representing the best solutions and its decrease in the case of less optimal paths. Among the advantages of pheromone trail evaporation, we can see, thus avoid too fast convergence of the algorithm, discarding not optimal paths and allowing execute the exploration of new paths in the

search space. From this equation, details and update rate of the pheromone are discussed below, (4):

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

Rate trace update of pheromones trails

Where m is the number of arbitrarily defined ants, ρ is the evaporation rate of pheromone trail that can take a value between zero and one, $\Delta\tau_{ij}^k(t)$ is the amount of pheromone left by ant k in each ij path for a time.

The decrease or evaporation of the amount of pheromone function of time for each cycle, 5:

$$\Delta\tau_{ij}^k(t) = \left\{ \frac{1}{L^k(t)} \right\} \quad (5)$$

Pheromone evaporation rate

Where $L^k(t)$ indicates the total length of the path taken for a given ant. The amount of pheromone adjusted by the rate of evaporation, can be seen as a mechanism for cooperative group memory and long-term operating to influence future decisions of the ants in the paths that should be explored, [14].

2.6. Stop criterion of the algorithm

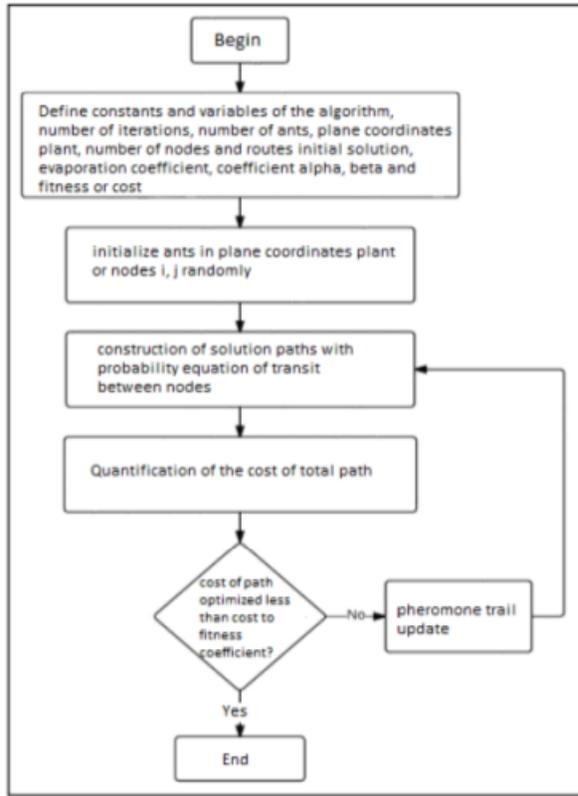
The Iteration of algorithm, using as criteria a maximum number of cycles and the cost coefficient or fitness, once this value is reached, the process ends, but not all cases can ensure that this solution is reached, but if it can achieve an optimized result.

3. Results

The ACO implemented algorithm is presented below, Figure 1:

- Initialize the pheromone trail between nodes or workstations of the layout for each of the paths.
- For all the ants to create a solution using pheromones.
- Evaporating the pheromone trail of every path in a certain amount for all solutions.
- Quantify the cost of the paths and identify the most optimal.
- Update the pheromone trail for each created solution.
- If the stop criterion is not satisfied, then the algorithm needs to go back to step b.
- If satisfaction was fulfilled.

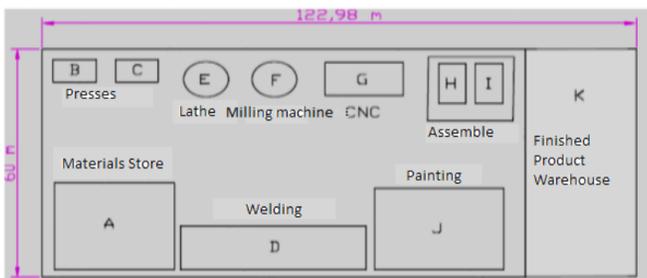
Figure 1: ACO algorithm Flowchart. Sources: [10–13].



Source: own.

Figure 2 shows, the plane of the floor of a machine shop, on which the path optimization is performed on an area of 123×60 meters, the nodes or sections that form are also indicated and the path must pass by them. Previously proposed algorithm was implemented as an m- file in Matlab script, the spatial coordinates (x, y) location of each station or node is entered as a vectors (x, y) .

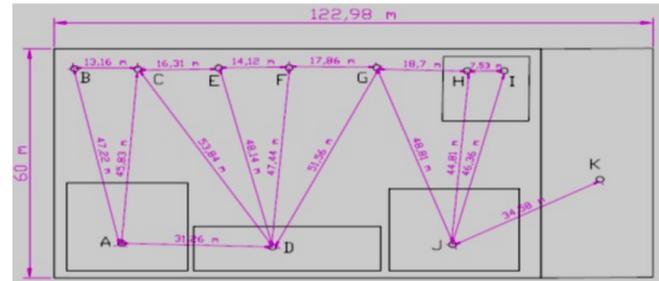
Figure 2: Coding sections and workstations on the layout.



Source: own.

The total area where the plant is distributed is 7,380 square meters. The total amount of monthly production is 1,000 pieces and each piece currently covers a distance of 558,34 m to be finished.

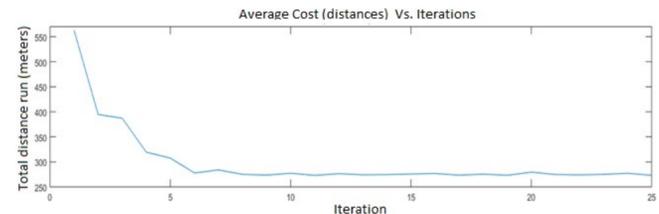
Figure 3: Initial paths distances in meters.



Source: own.

The assumption on which the simulation was developed, was determined by the following considerations, the algorithm ends when 25 iterations or fitness value or less than the desired cost are achieved. The minimum number of ants should be at least equal to the number of nodes or workstations, which in this case was taken in 200. The weight of the pheromone information at the beginning was taken arbitrarily like $\alpha = 0,9$, the relevance of heuristic value $\beta = 4,25$, the evaporation parameter $\rho = 0,01$, the probability of best path at start $q_0 = 0,9$. The initial value of pheromone τ_0 for each path is initialized randomly in a range from 0.1 to 0.5. The lower value of pheromone allowed in the algorithm must be a constant low value (0.001) to prevent the algorithm convergence in a local solution. The number of the best solution groups was assumed to equal to 25% of the amount of ants.

Figure 4: Cost curve optimization according to the iterations performed.



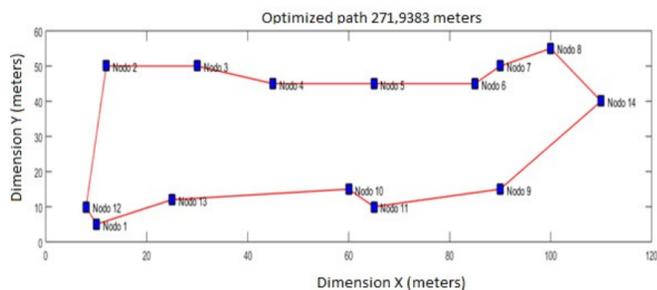
Source: own.

A journey is assigned to each ant k , from node i to node j , applying the formula of transitional probability. Ants repeat this process, until they reach all the number

of nodes initially set. Progressively evaluating the cost associated with each path to find the optimum value of cost or total distance in terms of iterations performed, see Figure 4.

After 25 iterations of the algorithm, the best solution found corresponded to the combination of Figure 5, with an associated cost of 271,93 meters. Compared with the initial configuration of the path that was at a value of 558,34 meters.

Figure 5: Optimized Tour.



Source: own.

4. Discussion

The costs associated with the layout management planning, depend among many aspects of the ability to optimize time, flow and handling of materials until a finished product. Under that premise, the scope of the results, evidenced by a final strategy for saving path but also the size optimization of material flow meter.

The unit costs are defined like paths in meters, to estimate the margins of the cost by path, so as to define production costs for each alternative. In each of the scenarios, it also minimizes the total production time. After completed the number of iterations for each simulation with the proposed scenarios, the cost calculations were performed to evaluate the best option and make decisions: the optimal solution from the point of view of operating costs.

5. Conclusions

The first consideration to create an ACO algorithm is to define a model or representation of the problem. You must define the objective function, taking into account, as to achieve greater fitness and really a better solution for the problem is given, [15–24].

The initial value of the path without optimization, starts from an initial value of 558,34 meters, with the ACO algorithm optimized a value of 271,93 meters was obtained as the final value. The ACO works particularly well solving combinatorial problems whose feasible solution space is too large to carry out a comprehensive search of reasonable time, as in this case optimization of paths and flows to the plant layout.

It was evident, the potential of the ACO in its implementation in logistic problems, as in the case of path planning, with good results as optimization method facilitating decision-making in this field.

References

- [1] R. H. Ballou. “Logística Administración de la Cadena de Suministro”. Quinta Edición, México: Pearson educación, págs., 14 y 64, 2004
- [2] K. Suzuki and M. Yokoo. “Secure Combinatorial Auctions by Dynamic Programming with Polynomial Secret Sharing”. Harvard School of Engineering and Applied Sciences. October 2014. [Online], available: <http://www.eecs.harvard.edu/cat/cs/diss/clock/suzuki02secure.pdf>
- [3] R. Bellman & S. Dreyfus. “Applied Dynamic Programming”. Princeton University Press, 1962.
- [4] W.L. Winston. “Investigación de Operaciones. Aplicaciones y Algoritmos”. Editorial Iberoamérica, 1994.
- [5] V. Maniezzo, L. M. Gambardella, and F. De Luigi, “Ant Colony Optimization,” Optimization Techniques in Engineering. SPRINGER-VERLAG, vol. 141, pp. 101–117, 2004.
- [6] A. Colomi, M. Dorigo, V. Maniezzo, and others, “Distributed optimization by ant colonies,” in Proceedings of the first European conference on artificial life, vol. 142, pp. 134–142, 1991.
- [7] M. Dorigo, V. Maniezzo, and A. Colomi, “The ant system: An autocatalytic optimizing process,” TR91-016, Politecnico di Milano, 1991
- [8] M.M. Flood. “The traveling salesman problem”. *Operations Research*, vol. 4, no. 1, Feb. 1956.
- [9] L. Kang and W. Cao, “An Improved Genetic & Ant Colony Optimization Algorithm for Travelling Salesman Problem,” in 2010 International Symposium on Information Science and Engineering (ISISE), pp. 498–502, 2010.

- [10] T. Stützle and M. Dorigo, "ACO algorithms for the traveling salesman problem," *Evolutionary Algorithms in Engineering and Computer Science*, pp. 163-183, 1999.
- [11] C. Blum, "Ant colony optimization: Introduction and recent trends". *Physics of Life Reviews*, vol. 2, no. 4, pp. 353-373, 2005.
- [12] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no 1, 1997, pp. 53-66.
- [13] Li, K., Kang, K., Zhang, W., & Li, B, "Comparative analysis of genetic algorithm and ant colony algorithm on solving traveling salesman problem", In: *Proceedings of the IEEE International Workshop on Semantic Computing and Systems*, pp.72-75, 2008.
- [14] G. Dantzig, R. Fulkerson, S. Johnson, "Solution of a Large-Scale Traveling-Salesman Problem" *Journal of the Operations Research Society of America*, vol. 2, no. 4 (Nov., 1954),
- [15] M. Nourelfath, N. Nahas, and B. Montreuil, "Coupling ant colony optimization and the extended great deluge algorithm for the discrete facility layout problem," *Engineering Optimization*, vol. 39, no 8, pp. 393-410, 2007.
- [16] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243-278, 2005.
- [17] Y. Hani, L. Amodeo, F. Yalaoui, H. Chen, "Ant colony optimization for solving an industrial layout problem," *European Journal of Operational Research*, vol. 183, no 2, pp. 633- 642, 2007.
- [18] K. Li, L. Kang, W. Zhang, and B. Li, "Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem," in *IEEE International Workshop on Semantic Computing and Systems*, 2008. WSCS '08, pp. 72-75, 2008.
- [19] R. Contreras. "Modelo de optimización en la generación de plantas industriales, considerando las actividades de mantenimiento y las condiciones ambientales mediante el uso de la metodología de los algoritmos genéticos". Tesis para optar el título de doctor. Universidad Politécnica de Valencia, pp. 81-281, 2010.
- [20] J. González, I. Rojas, H. Pomares, M. Salmerón and J.J. Merelo "Web newspaper layout optimization using simulated annealing." *IEEE Transactions on System, Man, and Cybernetics*, Part B, vol 32, no 5, pp. 686-691, 2002.
- [21] M. Dorzán, E. Gagliardi, M. Leguizamón, M. Taranilla. "Algoritmos ACO aplicados a problemas geométricos de optimización". Proyecto UPM AL09-PAC12 y Proyecto Tecnologías Avanzadas de Bases de Datos Universidad Nacional de San Luis, Argentina, 1- 2, 2009.
- [22] R. Martí. "Procedimientos metaheurísticos en optimización combinatoria". Universidad de Valencia. Departament d'Estadística i Investigació Operativa. Facultat de Matemàtiques, pp. 2-9, 2003.
- [23] M. A. Muñoz, J.A. López y E.F. Caicedo. "Inteligencia de enjambres: sociedades para la solución de problemas (una revisión)". *Revista Ingeniería e Investigación* vol. 28 no. 2, pp. 119-130, agosto de 2008.
- [24] ACO "Public Software" July 2015, [Online] Available in: <http://iridia.ulb.ac.be/mdorigo/ACO/aco-code/public-software.html>