International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

To cite this article: Trung Minh Ngo (2023). BUSINESS APPLICATIONS OF OPTIMISATION THEORY: ANT COLONY OPTIMIZATION AND SELECTED APPLICATIONS IN MANUFACTURING, International Journal of Research in Commerce and Management Studies (IJRCMS) 5 (4): 183-193 Article No. 227 Sub Id 420

BUSINESS APPLICATIONS OF OPTIMISATION THEORY: ANT COLONY OPTIMIZATION AND SELECTED APPLICATIONS IN MANUFACTURING

Trung Minh Ngo

Faculty of Business and Economics
University of Pecs, Pecs 7624, Hungary

DOI: https://doi.org/10.38193/IJRCMS.2023.5412

ABSTRACT

Ant Colony Optimization is a metaheuristic that was developed in the early 90s to solve optimization problems and finding for them good approximate solutions if not perfect ones, the developers of ant colony optimization algorithms took inspiration from the foraging behavior of ants in ant colonies, where although many ant species are close to blind, they have a surprisingly efficient technique in finding the shortest paths from their colony towards food sources and back. The first section of this paper introduces ant colony optimization origins and ideas, and the section after that discusses the main algorithms of ant colony optimization. The next chapter will include successful manufacturing applications for the algorithms. The last section will summarize the paper and discuss the findings. The main purpose of this paper is to discuss the ant colony optimization function and highlight how it has applications not only in theory but rather very consequential applications in business, namely, the world of manufacturing.

KEYWORDS: Optimisation Theory, Ant Colony Optimization, Flexible Manufacturing, Virtual Cellular Manufacturing

I. INTRODUCTION

Optimization problems are important for both theoretical fields, and for organizations in the real world, including the world of business. Examples of realistic optimization problems include vehicle routing problems, timetabling, data mining, cell placement in circuit design, communication network design, industrial problems, scheduling problems and many more of such nature (Blum, 2005).

The mentioned optimization problems are classified as combinatorial optimization (CO) problems. This class of optimization problems have “a finite set of objects (or search space) S and an objective function $f : S \rightarrow R+$ that assigns a positive cost value to each of the object’s $s \in S$. The goal is to find an object of minimal cost value” (Blum, 2005).

International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

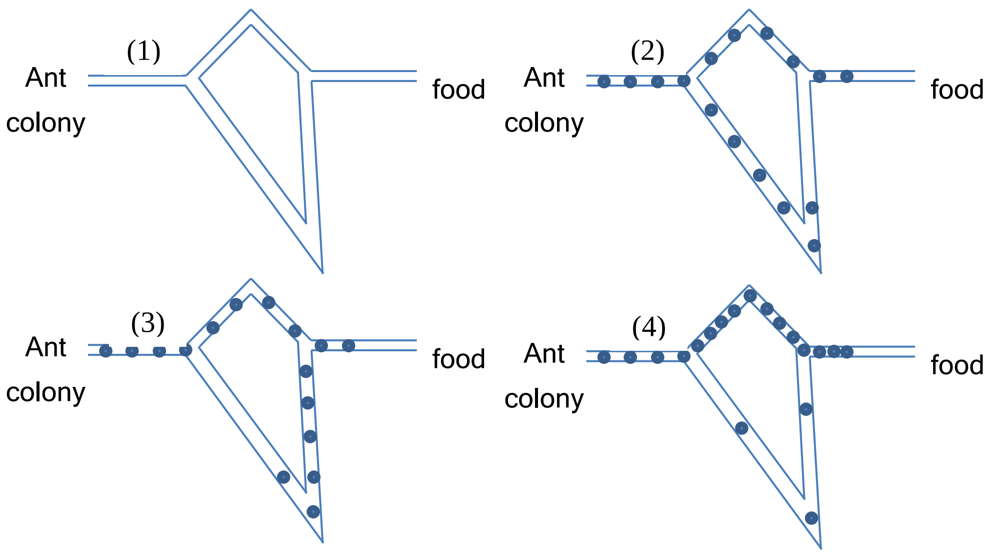
The unique factors of CO problems lies in that some of the information about the problem data is not known. In addition, assumptions are made about the knowledge of the probability distribution (Bianchi et al., 2008).

Some CO problems are easier to solve than others, and some CO problems are very difficult to solve. Indeed, CO problems that are classified NP-hard can be near impossible to solve with classical approaches due to computational feasibility (Dorigo and Stützle, 2019).

Because of the complexity of some CO problems, a classical approach of finding the optimal solution can take computational exponential time in the worst-case scenario (Bianchi et al., 2008). This gave rise to metaheuristic approaches to solve these kinds of problems in which having the optimal solution that can take extraordinary computational time to solve is abandoned, in return for having good solutions that may even be near optimal in a much shorter time span (Blum, 2005).

This paper's focus is Ant Colony Optimization (ACO), which is a metaheuristic that was developed for solving CO problems. The developers of ACO took inspiration from the behaviour of ants and ant colonies, specifically, the ants foraging behavior. Because of ants' social and collaborative interaction, they can find the shortest paths between their nest and food. Ants that leave their nest in search for food leave in their trail an odorous substance called pheromones. When there is no pheromone, ants move randomly, but then they start to move in paths marked with pheromone by each other. Paths that obtain more pheromone are more desirable for ants to follow. This inclination helps ants learn and then follow the shortest paths to food and back to the colony (Cordon et al., 2002; Dorigo et al., 2006). The collaborative interaction can be seen in figure (1).

Figure (1):



Based on a figure in Cordon et al. (2002) and Blums (2005) papers, the figure indicates how the ant colony starts to achieve the shortest path to food.

Using this biological example of ant communication and collaboration, ACO uses indirect communication between agents, which are called artificial ants that communicate through artificial pheromone paths (Dorigo and Stützle, 2019).

The ACO uses artificial ants by considering heuristic information about the problem instance and using the artificial pheromone paths. The artificial pheromone paths in reality are numerical information, which is then used by the artificial ants to probabilistically create solutions to the problem.

This allows the ACO to explore a vast array of solutions and, guided by the artificial pheromone, the artificial ants are led to the most adequate solutions. Moreover, the applications of ACO in reality are huge, because it can basically be used for any discrete optimization problem where a solution construction mechanism could be devised (Dorigo and Stützle, 2019). The next chapter will be about the main ACO algorithms.

II. The Main Ant Colony Optimization Algorithms

This section will include the three main ACO algorithms (Dorigo et al., 2006), the Ant System (AS), the *Max-Min* Ant System (MMAS) and the Ant Colony System (ACS). AS was the first algorithm. It is distinct in that at each iteration, the artificial ants (m) that have attempted to build solutions in an iteration update the artificial pheromone values at each iteration. The pheromone (τ_{ij}), is updated as follows in connection with the edge cities i and j (Dorigo et al., 2006):

$$\mathbf{t}_{ij} \leftarrow (1 - \rho) \,.\, \tau_{ij} + \sum (k{=}1)^{m} \ \Delta \ \mathbf{t}_{ij}{}^{k}$$

The ρ here represents the "evaporation" of the artificial pheromone, m is the number of artificial ants, and $\Delta \ \tau_{ij}{}^{k}$ is the amount of artificial pheromone laid by k.

$$\Delta \ \mathbf{t}_{ij}{}^{k} = \begin{cases} Q/L_k \\ 0 \end{cases}$$

**Q/L_k if the artificial ant used city edges i and j in its path, otherwise 0.
**Q here is the constant and L_k in the length of the path.

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

When constructing possible solutions, ants select to visit the next city by a stochastic solution mechanism. When *k* visited city *i* and created a partial solution S ^{p}, the probability of visiting *j* is going to be:

$$p_{ij}^k = \begin{cases} (t_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}) / (\sum c_{il} \in N\ (s^p) t_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \\ 0 \end{cases}$$

**If $c_{il} \in N\ (s^p)$, otherwise 0

N (S^p) here is the array of feasible components which are the city edges (*i*, *l*) where *l* is a city that *k* has not visited yet. And α and β are parameters that are there to control the importance of N_{ij} vs the pheromone trail, which is done by:

$$N_{ij} = \frac{1}{d_{ij}}$$

***d* here represents the distance between *i* and *j* (Dorigo et al., 2006).

The second algorithm that this paper will take mention of is the Min-Max Ant System. The MMAS was an improvement on the original AS, mainly in that the best artificial ants are the ones that update the pheromone paths, and that the values of the artificial pheromones are bound. Which is applied in the following equation:

$$t_{ij} \leftarrow \left[(1-\rho).t_{ij} + \Delta\ t_{ij}^{best}\right]_{t\,min}^{t\,max}$$

Where the t_{max} and t min are imposed on the pheromone as the upper and lower boundaries, and you can define the X_b^a operator as:

$$f(x) = \begin{cases} a, & if\ x > a, \\ b, & if\ x < b, \\ x, & otherwise. \end{cases}$$

And $\Delta\ t_{ij}^{best}$ is:

$$\Delta\ t_{ij}^{best} = \begin{cases} 1/L_{best}, & if\ (i\ ,j) belong\ to\ the\ best\ path, \\ 0, & otherwise. \end{cases}$$

Here, L_{best} is the best ant's path full length, whilst t_{min} and t_{max} are usually acquired empirically

(Dorigo et al., 2006).

Last of the main ACO algorithms is the previously mentioned ACS. This function proposed a "*local pheromone update*" that each artificial ant executes after every single construction step as follows (Dorigo et al., 2006):

$$t_{ij} = (1 - \varphi).t_{ij} + \varphi.t_0$$

Here $\varphi \in (0,1]$ is the coefficient of the pheromone decay, and that initial value given to pheromone is t_0. The local update was developed in order to achieve different results rather than having identical solution in the same iterations. This is achieved because the artificial ants in this update encourage the following artificial ants to create different results by adopting different paths (Dorigo et al., 2006).

Additionally, there is an offline pheromone update by the end of iterations that is performed by one artificial ant. The update formula is as follows:

$$t_{ij} = \begin{cases} (1-\rho).t_{ij} + \rho.\Delta t_{ij}, & if\ (i,j) belong\ to\ the\ best\ path, \\ t_{ij}, & otherwise. \end{cases}$$

The last relevant distinction between the original AS and the ACS lies in the decision rule used by the artificial ants in the process of construction solutions; the probability of an artificial ant moving from *i* to *j* depends on *q*, a random variable distributed uniformly over [0,1] and a parameter q_0; if q <= q_0, then *j* = argmax $c_{il} \in N\ (s^p)\{t_{il}\ \eta_{il}^b\}$, otherwise, the previous equation is used:

$$p_{ij}^k = \begin{cases} (t_{ij}^{\alpha} \ .\eta_{ij}^{\beta}) / (\sum c_{il} \in N\ (s^p) t_{ij}^{\alpha} \ .\eta_{ij}^{\beta} \\ 0 \end{cases}$$

After discussing the main algorithms of Ant Colony Optimization, the next chapter will discuss the real business applications that have benefited from this optimization solution.

III. Applications of Ant Colony Optimization in real business settings

1. Minimum Tool Switching Problem in Flexible Manufacturing systems

The first application this paper discusses is a problem that manufacturers face in tool switching, in a flexible manufacturing system. The problem that was introduced by Konak and Kulturel-Konak (2007) was that in this case, the manufacturers had flexible manufacturing systems where parts are processed on CNC machines with tools that are loaded depending on the desired requirements of the CNC machine.

In case the parts that were processes needed more tools than the total tool magazine capacity of the CNC machine, the manufacturer will inevitably have to stop the process and load the required tools.

International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

Which in turn lengthens the manufacturing process. This kind of problem is well recognized the manufacturing literature, and therefore needed a solution (Konak and Kulturel-Konak, 2007).

Konak and Kulturel-Konak's (2007) proposed solution using ACO was as follows. The researchers first defined the pheromone path as the desirability (for efficiency purposes) of the CNC machine processing type *i* and type *j* in the same "*tool switching instant*". By defining the i and j in this way, the pheromone matrix is symmetric where t (*i*,*j*) = t (*j*,*i*).

The researchers then went on to build a feasible solution as the following:
Let $\boldsymbol{\Pi} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_m}\}$ represent a solution. With *m* being the number of tool changing accounts. S_i is the set of parts in the i^{th} account. Here, each artificial ant starts with an empty solution $\boldsymbol{\Pi} = \boldsymbol{s_1}$ and $S_1 = \emptyset$. After that, the solution randomly commits parts to the machine until full capacity of the tool magazine is established, then a new instant or account is established, $\boldsymbol{\Pi} = \{\boldsymbol{s_1}, \boldsymbol{s_2}\}$ and $S_2 = \emptyset$.
In the solution, S_m represent the current account and $A(S_m)$ the array of parts that are possible to designate to S_m. The array of $A(S_m)$ includes the parts that have not been designated to an account yet, and that do not breach the tool magazine capacity load if designated to S_m. In this case, the probability of designating a part is given by the following formula:

$$\boldsymbol{p(i, s_m)} = \begin{cases} \dfrac{\boldsymbol{t(i)\eta(i)^{\beta}}}{\sum_{j \in s_m} \boldsymbol{t(j)\eta(j)^{\beta}}}, & \boldsymbol{if\ i\ \in A(s_m)} \\ \boldsymbol{0}, & \boldsymbol{otherwise} \end{cases}$$

Where: t_i is the total pheromone trail part *i* and η(*i*) is the distinct information of the problem. Here, designating parts to empty of new accounts, the parts that have a higher tool requirement will be favourable to the parts that have a lower tool requirement. The total pheromone trail $t(_i)$ depends on whether current instant S_m is empty or not as given in the next equation.

$$\boldsymbol{\eta_{(i)} = C + 1 - |s(s_{m} \smile i)|}$$

$\boldsymbol{|s(s_m \smile i)|}$ here is the array of tools that are required by the array of parts in S_m and *i* together (Konak and Kulturel-Konak, 2007). The researchers then started and developed the pheromone path function to suit their purposes, and finally arrived at the following overall algorithm:

"t ←0
initialize t (i, j)
While (t < t $_{max}$) do
Randomly create n $_{pop}$ solutions using probabilities in

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

$$p(i, s_m) = \begin{cases} \dfrac{t(i)\eta(i)^{\beta}}{\sum_{j \in s_m} t(j)\eta(j)^{\beta}}, & if\ i \in A(s_m) \\ 0, & otherwise \end{cases}$$

Sort the population
Identify and delete duplicate solutions
Update best-solution-so-far
Update r(i, j) using the best solutions of the cycle
t ←t + 1
End
Return best-solution-so-far."

This approach was the first of its kind in use for tool switching problem of flexible manufacturing systems. The results it showed when applied were excellent as the method found optimal solutions for the majority of the test problems. The results therefore showed great promise (Konak and Kulturel-Konak, 2007).

2. Virtual Cellular Manufacturing problem

The virtual cellular manufacturing problem is the second real business problem that researchers found better solutions for using ACO. Cellular manufacturing is process that is a subset of just-in-time manufacturing. In this approach to manufacturing, units that have similar design or manufacturing attributes are grouped together into manufacturing cells, which make material flow simple in these manufacturing cells in order to reduce excess time (Mak et al., 2007).

The problem with the process of cellular manufacturing is that the manufacturing cells suffer from an unbalanced workload between machines and also suffers from low utilizations of some machines. Furthermore, because of the setup, cellular manufacturing also suffers a higher cost of operations and maintenance and complicated material flow schemes between the manufacturing cells. In order to overcome these issues, virtual cellular manufacturing was developed, in which the manufacturing cells are not identified physically in one production space, rather, they are identified virtually as data files in a virtual cell controller, which receives the manufacturing order, and subsequently choose a group of workstations on the production floor to form a virtual manufacturing cell to fill out the exact requirement for that specific manufacturing order (Mak et al., 2007).

Thus, an ACO model was proposed by Mak et al. (2007), in which the virtual cell controller can produce manufacturing schedules for the machines in a way that minimizes travel time for raw materials, semi-final products and assembled products. The following was the researchers' suggested model steps to solve this problem:

- *"Initialize ACO parameters and pheromone trails.*
- *Send ants out to visit all nodes involved and construct their respective tours.*
- *Evaluate the tours constructed.*
- *Update pheromone trails.*
- *If the termination conditions are not satisfied, go back to step 2; otherwise, terminate".*

In the path construction phase, the following algorithm was applied:

$$\boldsymbol{j} = \begin{cases} \arg \boldsymbol{max}_{\boldsymbol{u} \in \boldsymbol{s_k(i)}} \{[\boldsymbol{t_{iu}}]^{\alpha} [\boldsymbol{\eta_{iu}}]^{\beta}\}, & \boldsymbol{if\ q <= q_0} \\ \boldsymbol{j}, & \boldsymbol{if\ q > q_0} \end{cases}$$

Here, $\boldsymbol{\eta_{iu}}$ is a heuristic, it equals to inverse of the length $\boldsymbol{d_{iu}}$ from node *i* to node *u*. $\boldsymbol{t_{iu}}$ represents the amount of pheromone in the path. And α and β are parameters used to control the weight of the pheromone. $\boldsymbol{s_k(i)}$ contains the untouched nodes for the artificial ant *k* which is at node *i*. *q* is the random number which is uniformly distributed [0,1].

And $q_0 (q_0 \in [0,1])$ is a parameter set to determine the importance between exploitation of the paths and exploration of the ants. $\boldsymbol{j}$ is the randomly chosen node from the list $\boldsymbol{s_k(i)}$.
The probability that ant *k* chooses to travel from *i* to *j* is represented by the following formula:

$$\boldsymbol{p_{ij}^k} = \begin{cases} \dfrac{[\boldsymbol{t_{ij}}]^{\alpha} [\boldsymbol{\eta_{ij}}]^{\beta}}{\sum_{\boldsymbol{u} \in \boldsymbol{s_k(i)}} \left\{[\boldsymbol{t_{ij}}]^{\alpha} [\boldsymbol{\eta_{ij}}]^{\beta}\right\}}, & \boldsymbol{if\ j \in s_k(i)} \\ \boldsymbol{0}, & \boldsymbol{if\ j \notin s_k(i)} \end{cases}$$

The researchers also established local and global updates for the pheromone paths, and proceeded to detail the procedure implement it on multiple experiments as well as an actual manufacturing facility in China that uses virtual cellular manufacturing. In both instances, the ACO showed better performance in computational time required, utilization of machines rate and the completion time of manufacturing tasks (Mak et al., 2007).

3. The Reconfigurable Manufacturing System problem

Similar to the two previous problems, this problem addresses the issue of efficiency and minimizing capital cost of the system. Reconfigurable manufacturing systems are systems that are created in way which allows for swift change, both in its structure and components, be it software or hardware, in order to meet new market demand or comply with internal requirements (Koren et al., 1999). The

International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

structure of reconfigurable manufacturing systems constitutes of CNC machines and reconfigurable machine tools (Maniraj et al.,2015).

Using ACO, Maniraj et al. (2015) aimed at achieving efficiency and cost reduction in processes of reconfigurable manufacturing systems by finding optimal machine assignment. They used ACO formula with minor modifications to suit their requirements, and the formula they used looked as follows:

$$\boldsymbol{CC} = \sum_{s=1}^{N} (\boldsymbol{n_s}.\boldsymbol{cm_s}) \,.\, [\mathbf{1} - ((\mathbf{1} - \boldsymbol{D})^{\boldsymbol{t}}.\boldsymbol{PWF})]$$

Here, CC indicates capital cost, and $\boldsymbol{PWF} = \mathbf{1}/(\mathbf{1} + \boldsymbol{I})^{\boldsymbol{t}}$ is the present worth factor. $\boldsymbol{cm_s}$ here is the cost of machines used in stage *S*. $\boldsymbol{n_s}$ is the number of operation cluster setup. And $\boldsymbol{D}$ is the rate of depreciation in the machines (Maniraj et al.,2015).

After creating Feasible Operation Clusters, the researchers used ACO in order to reach optimal machine assignment with the following steps:

"Step 1. Initial population generation with a number of ants and their corresponding CC values. An ant is considered as a permutation of random sequence formed with the available operation clusters without any repetition.
Step 2. Finding pheromone matrix values for the selected population.
Step 3. Updating pheromone matrix values for every iteration.
Step 4. Finding next set of orders using pheromone matrix. Here randomly generated position sequence and normalized values helps in finding normalized vector. Furthermore, the lowest value of the normalized vector is used for assigning the position for the next of sequence in ant population. This creates a new set of population for the next generation.
Step 5. Find out the corresponding CC values for the newly generated population.
Step 6. Repeat the Steps 3–5 until it reaches the maximum number of generations."

By using the novel approach of ACO, the researchers were able to fulfil their goal of finding optimal machine assignment a reconfigurable manufacturing system guaranteeing reduction of capital cost and further efficiency in the manufacturing system (Maniraj et al.,2015).

IV. DISCUSSION

This paper discussed Ant Colony Optimization theory, that main ACO algorithms and further discussed some uses in manufacturing that were established in previous literature, namely the minimum tool switching problem in flexible manufacturing systems (Konak and Kulturel-Konak,

International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 5, No. 04 July-Aug; 2023 Page. No. 183-193

2007), the virtual cellular manufacturing problem (Mak et al., 2007) and the reconfigurable manufacturing system problem (Maniraj et al.,2015).

It should also be mentioned that due to length constraints of writing this paper, it only mentioned three cases where ACO was used, but they is a wealth of scholarly articles that focus on ACO in manufacturing problems such as the bottleneck station scheduling problem for assembly and test manufacturing that was done by Song et al. (2007), or robot path integration using ACO (Tewolde and Weihua Sheng, 2008), or automated manufacturing processes plans (Tiwari et al., 2006) to name a few.

The array of uses for ACO is enormous and it has deep importance in many more fields. This paper only highlighted a few uses in a very narrow field as per to meet the requirements, but the ACO algorithm's applications are almost obscenely underrepresented in this paper. Optimization problems in general are a necessity not only in theory, but in practice as well, unless we have infinite computational time and infinite number of super computers.

REFERENCES

- Bianchi, L., Dorigo, M., Gambardella, L.M. and Gutjahr, W.J. (2008). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2), pp.239–287.
- Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4), pp.353–373.
- Cordon, O., Herrera, F. and Stutzle, T. (2002). a review of ant colony optimization metaheuristic: basis, models and new trends. *Mathware & Soft Computing*, 9(2–3).
- Dorigo, M., Birattari, M. and Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), pp.28–39.
- Dorigo, M. and Stützle, T. (2019). Ant Colony Optimization: Overview and Recent Advances. *Handbook of Metaheuristics*, pp.311–351.
- Konak, A. and Kulturel-Konak, S. (2007). An Ant Colony Optimization Approach to the Minimum Tool Switching Instant Problem in Flexible Manufacturing System. *2007 IEEE Symposium on Computational Intelligence in Scheduling*.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H., 1999. Reconfigurable manufacturing systems. *CIRP annals*, 48(2), pp.527-540.
- Mak, K.L., Peng, P., Wang, X.X. and Lau, T.L. (2007). An ant colony optimization algorithm for scheduling virtual cellular manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 20(6), pp.524–537.
- Maniraj, M., Pakkirisamy, V. and Jeyapaul, R. (2015). An ant colony optimization–based

approach for a single-product flow-line reconfigurable manufacturing systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(7), pp.1229–1236.

- Song, Y., Zhang, M.T., Yi, J., Zhang, L. and Zheng, L. (2007). Bottleneck Station Scheduling in Semiconductor Assembly and Test Manufacturing Using Ant Colony Optimization. *IEEE Transactions on Automation Science and Engineering*, 4(4), pp.569–578.
- Tewolde, G.S. and Weihua Sheng (2008). Robot Path Integration in Manufacturing Processes: Genetic Algorithm Versus Ant Colony Optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38(2), pp.278–287.
- Tiwari, M.K., Dashora, Y., Kumar, S. and Shankar, R. (2006). Ant colony optimization to select the best process plan in an automated manufacturing environment. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(9), pp.1457–1472.