Team project

OPTIMIZATION

(MECH-8290)

Submitted by:

Shubham Dipak Shinde [110128036]

Dhruvi Kirit Kumar Panchal [110142400]

Arth Dharmesh Gaudani [110141827]

Tirth Minesh Gandhi [110133166]

Submitted to:

Prof: Amr Shabaka

"Cost of quality and quality optimization in manufacturing using particle swarm optimization"

Table of Contents

Table Of Figures

Acknowledgement:

We would like to extend our heartfelt gratitude to our esteemed professor, Dr. Amr Shabaka, for your invaluable guidance and support throughout our fourth semester. It was an immense pleasure to learn from your vast knowledge and experience in optimization. your ability to motivate us in every class, combined with your use of real-life examples, made the subject not only engaging but also highly practical. The lectures, assignments, and class activities were a remarkable experience that left a lasting impact on our understanding and appreciation of optimization. As a group, we sincerely thank you for your dedication and inspiration, which significantly enriched our learning journey.

Introduction:

The introduction emphasizes how important quality management is to the manufacturing sector's performance. The policies, processes, and practices that guarantee products meet or beyond customer expectations while preserving effectiveness and cost-effectiveness are together referred to as quality management. Upholding high standards of quality is not just a desirable objective but also a necessary condition for businesses to prosper in a global marketplace that is becoming more and more competitive.[1]

The Cost of Quality (CoQ), which includes all costs spent to guarantee product excellence, is a crucial component of quality management. Organizations can balance cost effectiveness and quality enhancements by optimizing CoQ. Because higher quality lowers errors, rework, and unhappiness, this optimization may result in increased customer satisfaction. Furthermore, efficient Coq management enhances resource efficiency, reduces waste in manufacturing processes, and eventually increases profitability.

The study explores the different aspects of Tcoq, including internal failure, external failure, appraisal, and preventative costs. Proactive steps to prevent faults, like process enhancements and training, are part of prevention expenses. Inspections, testing, and audits to make sure product requirements are fulfilled are covered by appraisal charges. Defects found before delivery, like scrap or rework, result in internal failure costs. Defects discovered after delivery result in external failure costs, such as warranty claims, returns, and harm to one's reputation.

By looking at these elements, this study highlights how quality control may greatly increase total manufacturing efficiency when combined with cost optimization techniques. This analysis is further enhanced by the incorporation of cutting-edge approaches like Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), which offer practical insights for manufacturers aiming to satisfy quality standards while maintaining cost competitiveness in the global market.[2]

Importance of Quality Optimization in Manufacturing:

In order to secure market share, improve operational efficiency, and satisfy customers, quality optimization is essential. It creates a clear connection between end users' opinions and experiences and the caliber of the goods or services provided. Businesses that make quality optimization investments not only meet but frequently surpass consumer expectations, which fosters greater brand loyalty, repeat business, and favorable word-of-mouth referrals.

Organizations that place a high priority on quality management include strict standards into their manufacturing and service delivery procedures. Customers find items more appealing because of these standards, which guarantee consistency, dependability, and conformity to requirements. Businesses are positioned as industry leaders thanks to this attention to detail and dedication to quality, which also increases client trust. By concentrating on quality, businesses are able to stand out from the competition, command higher prices, draw in affluent clients, and increase their market share.

In sectors where accuracy and dependability are essential, the importance of quality optimization is even more apparent. Customers in the automotive manufacturing industry, for example, want cars that are not only highly effective but also safe, long-lasting, and fuel-efficient. Quality optimization is essential since any quality compromise could result in expensive recalls, harm to one's reputation, or even safety risks. Similar to this, in the quickly evolving electronics sector, goods like laptops, smartphones, and consumer electronics are evaluated on the basis of their functionality, robustness, and novel features. Businesses that fall short of these standards run the danger of losing their competitive advantage in a market where customer satisfaction is paramount.

Businesses can increase operational efficiency by optimizing their processes to eliminate waste, reduce errors, and simplify production by integrating quality management into their core operations. The strategic significance of quality optimization in promoting long-term corporate success is highlighted by the combined advantages of higher market share, enhanced customer happiness, and greater operational efficiency.[3]

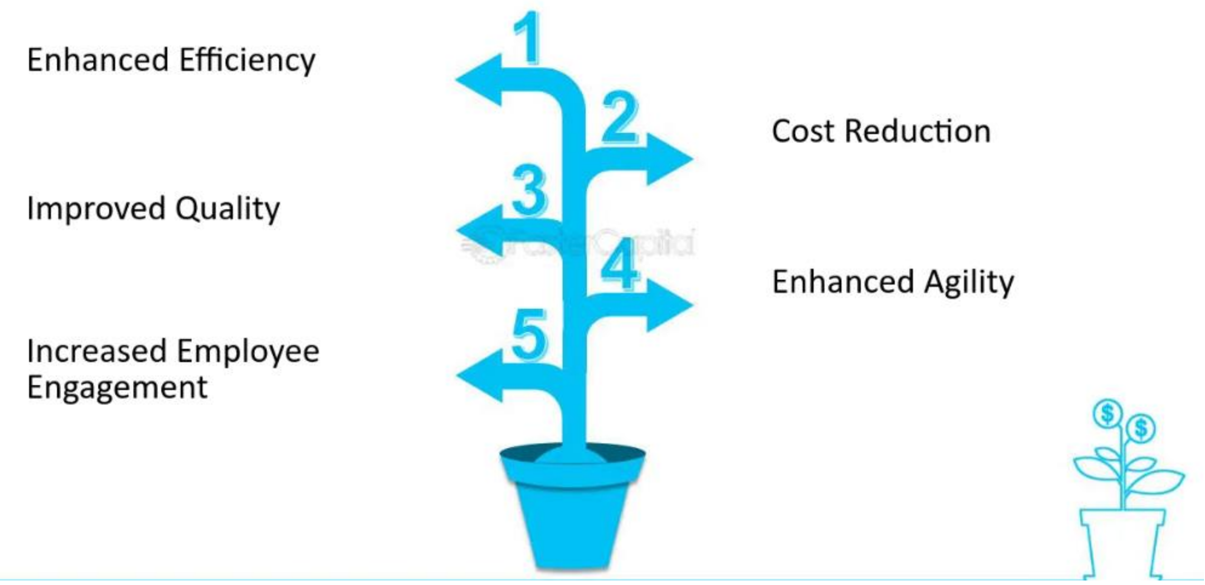

Figure 1 ***Importance of Quality***

Cost of Quality: Concepts and Analysis:

A thorough method for assessing and controlling the costs related to guaranteeing and preserving product quality is the Cost of Quality (CoQ) framework. It separates these expenses into four groups: internal failure costs, external failure costs, evaluation costs, and preventative costs. Every category has distinct actions and cost ramifications, representing a distinct facet of the quality management process.

Prevention costs are the proactive expenditures incurred to stop errors and defects in processes or products before they happen. The creation of strong quality management systems, process improvement projects to streamline processes, and staff training programs to increase skills are a few examples of preventative efforts. By emphasizing prevention, businesses can deal with possible problems at their source and reduce the possibility of errors and malfunctions. To avoid mistakes and save money on waste and rework, a manufacturing company might, for example, teach staff on quality standards or use modern production systems.

Appraisal costs are incurred when products are measured, tested, and audited to make sure they adhere to quality standards. Performance testing, quality audits, and inspections are typical appraisal procedures. For instance, cars in the automotive sector go through extensive testing to ensure that they meet performance standards and safety requirements. Even though evaluation fees are a substantial outlay of funds, they are necessary for early problem detection and guaranteeing that only superior items are delivered to clients.

Internal Failure Costs: These result from flaws found prior to the product being sent to the client. The expenses of fixing damaged goods, discarding useless products, and the inefficiencies brought on by production halts are a few examples. For example, the cost of fixing faulty circuit boards found during quality control inspections may be included in internal failure costs at an electronics manufacturing facility. Larger, more costly issues later in the supply chain can be avoided by addressing these shortcomings internally.

When faulty products are delivered to clients, the most harmful expenses are known as **external failure costs.** Reputational harm, product recalls, warranty claims, and legal liability are a few examples. For instance, if a product is recalled because of contamination, a food production company may incur external failure costs, which could lead to compensation claims and damage to the company's reputation. Because these expenses are frequently much higher than those in the other categories, preventative and evaluation efforts are crucial.

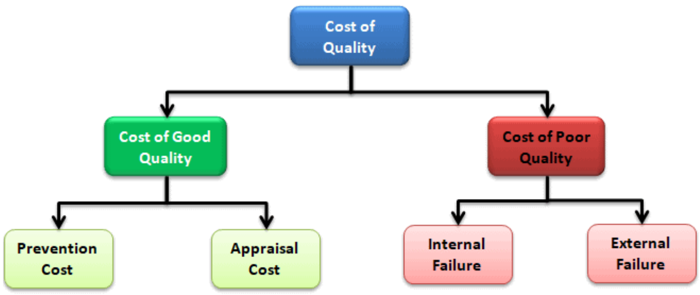

Figure 2 Cost of Quality

Research Objectives:

The study focuses on important issues with product quality and Cost of Quality (CoQ) optimization. It looks at how to find product elements that successfully strike a balance between price and quality. It also assesses methods for two scenarios: keeping costs around $22,000 to support a cost leadership plan and meeting high quality requirements (above 90%) in a product differentiation approach. These questions seek to offer practical advice for increasing productivity while achieving a range of corporate goals.

This study also shows the comparison between the optimization methods that the group chosen to provide a better solution or the best possible solution that is the most optimized at the moment. The problem is taken from the research paper and being analyzed thoroughly, the problem and its solutions, and then later optimized using the chosen method which would give better optimized results than the used method.

Therefore, there are plenty of possible methods to use for the optimization of a problem in real life situations. All the methods give best possible solution that leads to the optimized value of the problem. In this research paper chosen, it uses the ACO (Ant Colony Optimization) to prove the best possible solutions according to different situations of one problem stated in the paper. We also use the same problem given and chose to use the PSO (particle Swarm Optimization) method. The reason for choosing this particular method is that it gives more accurate form of optimized results and also uses the latest technology platform to avoid any errors or minimize the errors to converge a better optimized result.

Ant Colony Optimization (ACO)

ACO is therefore metaheuristic in the sense that the absolute optimum solution is not found, but good solutions practically close enough to the optimum are found. Real ants coordinate their activities through stigmergy, which is a form of indirect communication (Dorigo, 1992) [7]. Specifically, in the search for food, ants deposit chemicals along the path they travel which is recognized by other ants, and will increase the probability of the path being traveled by other ants of the colony. The chemical is called Pheromone (Stutzle, 1999) [8]. The fundamental components of ACO can be briefly categorized as:

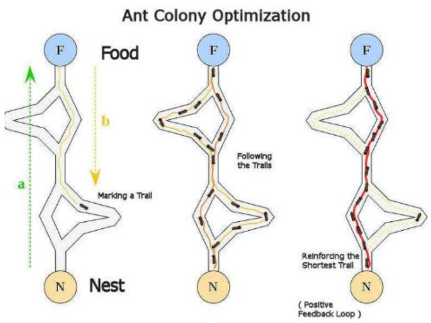

Figure 3 Ant colony optimization

I- Construct a graph of the problem. The resulting graph will have m columns, and each column has n nodes.

II- Define the objective function and the restrictions. The number of ants and the number of attempts for solving the problem are also specified in this step.

III- Move artificial ants on the graph in order to construct admissible solutions to the problem: In this step, artificial ants placed on the initial point start moving, randomly selecting a node on each consecutive column in order to build incremental solutions to the problem under consideration. The more ants placed on the graph, the more cross sections produced, and the higher the chances are that the best solution is approached. In selecting the nodes of a column to move to, the probability of an ant selecting the j-th node of the i-th column is described by the following relation: , , , i j i j i j p α α τ τ = ∑ (1)

$$p_{i,j} = \frac{\tau_{i,j}^{\alpha}}{\sum \tau_{i,j}^{\alpha}} \qquad (1)$$

In Eq.(1), τ i,j is the sum of the pheromone placed on node (i,j) from previous attempts. In the first attempt, all nodes have an equal pheromone of τ0 , and therefore in the first attempt, all nodes have an equal chance of being selected by the ants.

IV- Evaluate the solutions obtained by each ant in the first attempt: Once all the ants complete the first attempt, the objective function f is calculated for each ant. Next, pheromone is deposited along the trail which each ant has chosen in forming an incremental solution.
The amount of pheromone deposited on each node is reversely related to the objective function of the path being considered, i.e.,t = 1 /f. As the rule states, the lower the objective function of a path, the more pheromone will be deposited on the components of the path.

V- Update the pheromone value of each node in the graph: After calculating the pheromone value of every node for the present attempt, the updated pheromone value of each node is obtained through the following relation:

$$\tau_{i,j}(t+1) = (1-p)\tau_{i,j}(t) + \Delta\tau_{i,j}(t) \qquad (2)$$

In Eq.2, Δτi.j is the difference between the deposited pheromone in the present attempt and the previous attempt, τi.j is the updated pheromone value, and r is the evaporation index which takes a value between zero and one. Pheromone evaporation is a useful form of forgetting, preventing the algorithm from rapidly converging towards local optima. The term (1-r) thus determines how much of the pheromone accumulated from previous attempts is evaporated.

VI- Repeat steps III through IV in the next attempts in order to reach the optimum solution: in the next attempt, the decision making process of the artificial ants is no longer completely by chance; as stated by Eq.1, nodes with more pheromone have a higher chance of being selected by the ants. After each attempt, pheromone values are updated and some pheromone is evaporated. The combined action of pheromone deposit and evaporation enables a constant exploration of the search space towards a global optimum in ACO.

The above-mentioned steps form the fundamental framework of the ACO algorithm. Various improvements have been introduced to the original algorithm in recent years, aiming to make the search algorithm both more effective and more efficient. Accordingly, in addition to the ants system (AS) algorithm discussed above, three other algorithms have been more successful, and have been used in the present study: ranked ant system (ASrank), elite ant system (ASelite), maximum-minimum ant system (MMAS). The principle features of AS and MMAS which are applied in this research and elite ant system and ranked ant system are briefly discussed herein.

Ants System (AS): This is the simplest form of ACO first introduced by Dorigo et al. (1991) [9]. In AS, artificial ants choose their path according to the following probabilistic relation:

$$\rho_{ij}(k,t) = \frac{[\tau_{i,j}(k,t)]^{\alpha}[\eta_{i,j}(k,t)]^{\beta}}{\sum_{j=1}^{J}[\tau_{i,j}(k,t)]^{\alpha}[\eta_{i,j}(k,t)]^{\beta}} \qquad (3)$$

In which ρi,j(k,t) is the probability of selecting i-th node of the j-th column, by the k-th ant in the t-th attempt. ηi,j(k,t) in Eq.3 represents the heuristic information and the determination of its value is problem specific. In some problems, the value of ηi,j(k,t) is hard to determine, and is therefore omitted from equation. a and b in Eq.3 are constants which determine the role of pheromone and heuristic information in the artificial ants' decision-making process. If a>b, the role of pheromone is emphasized, and heuristic information has less effect on the decision of the ants.

$$\Delta\tau_{i,j}(t) = \sum_{k=1}^{m}\frac{Q}{f(s_k(t))}I_{S_k(t)}\{(i,j)\} \qquad (4)$$

In which m is the number of artificial ants, or the number of solutions produced; Q is a constant named the pheromone return index and its value depends on the amount of pheromone deposited; Sk (t) represents all the nodes which the k-th ant has chosen on the t-th attempt; ISk (t){(i , j)} is a coefficient which is either zero or one, depending respectively on whether the k-th ant has chosen the node (i, j) or not. In other words, ISk (t) ensures that only the nodes on which the k-th ant has moved to will be considered in depositing pheromone. It can be deduced from Eq.4 that in AS, solutions with a lower objective function will have more pheromone deposited, and vice versa. In Elitist Ants System this algorithm, more attention is focused on the elite ant of the colony. The elite ant is the one which has produced the best answer in all previous attempts. Specifically, in ASelite extra pheromone is deposited on the path which the elite ant has produced. The ants decide which node to move to using Eq.3.

The pheromone update rule in **ASelite** is:

$$\tau_{i,j}(t+1) = (1-p)\tau_{i,j}(t) + \Delta\tau_{i,j}(t) + \sigma\Delta\tau_{i,j}^{gb}(t) \qquad (5)$$

Whereas g b i j Δτ (t) is the extra pheromone deposited by the elite ant and is the weight of the extra pheromone. ASelite is an attempt to balance between exploration and exploitation in the algorithm. And Social science section 1077 Openly accessible at http://www.european-science.com in ranked ants' system algorithm, unlike the ASelite in which all ants participate in the pheromone update process, onlys-1elite ants which have created better solutions are chosen to update the pheromone of the paths they have chosen. In ASrank, following each attempt, the ants are lined up according to the solutions they have obtained, and pheromone update values are assigned to each ant, the most pheromone being assigned to the best solution and decreasing thereafter to the last ant in the line. Thus, the pheromone update rule in ASrank can be stated as:

$$\Delta\tau^{rank}{}_{i,j}(t) = \sum_{k=1}^{\sigma-1}(\sigma-k)\frac{Q}{f(S_k(t))}I_{S_k(t)}\{(i,j)\} \quad (6)$$

Minimum-Maximum Ants System (MMAS): Stutzle and Hoos (2000) [10] first reported the MMAS algorithm in a successful attempt to improve the efficiency of AS. The general structure of MMAS is similar to AS. However, only the path with the best solution in each attempt is chosen to deposit pheromone on its trail. In this way, the solution rapidly converges to the optimum. The danger always exists that the ants quickly move towards the first optimum solution achieved, before having the chance to explore other possibly better solutions in the search space. In order to prevent this from occurring, a restriction is placed on the minimum and maximum allowable net pheromone deposit on the trails, i.e., the deposited pheromone value is limited to [τmin, τmax]. Following each pheromone deposition step, all pheromone values are controlled to fit within the mentioned limit, and any node for which the pheromone value exceeds the limits is adjusted to the allowable limit. This is a way to promote the ants to explore new solutions in the search space. The maximum and minimum allowable pheromone values of the t-th attempt are calculated as:

$$\tau_{\max}(t) = \frac{1}{1-\rho}\frac{Q}{f\left(s^{gb}(t)\right)} \qquad (7)$$

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)\left(1-\sqrt[n]{p_{best}}\right)}{\left(NO_{avg}-1\right)\sqrt[n]{p_{best}}} \qquad (8)$$

Where f (S_{gb} (t)) is the value of the objective function up to the t-th attempt, Pbest is the probability of the ants choosing the best solution once again, NO_{avg} is the average of the number of decision choices in the decision points. It is noteworthy to mention that the initial pheromone value associated with the nodes, 0 τ is)(max τ t . Objective function is defined according to Eq.9:

$$F(x) = \frac{TCOQ - TCOQ_{\min}}{TCOQ_{\max} - TCOQ_{\min}} + \frac{Q_{\max} - Q}{Q_{\max} - Q_{\min}} \qquad (9)$$

Where TCOQ is the total cost of quality and Q is quality of each resource option for activities. Because the cost of quality and the quality have different module, by Eq.8 they become normalize. Now they can be comparable, added or subtracted. In this case study minimum cost of quality ($TCOQ_{min}$) is \$19,905, maximum cost ($TCOQ_{max}$) is \$26,770, minimum quality (Q_{min}) 79.95 and maximum quality (Q_{max}) is 95.32.

The total cost of quality for each option for compatible alternatives is calculated according to Eq.10:

$$TCOQ = \sum_{n=1}^{9} PC_n + \sum_{n=1}^{9} AC_n + \sum_{n=1}^{9} IFC_n + \sum_{n=1}^{9} EFC_n \qquad (10)$$

Where PCn is the prevention cost of each option for compatible alternatives, ACn is the appraisal cost of each option for compatible alternatives; IFCn is the internal failure cost of each option for compatible alternatives and EFCn is the external failure cost of each option for compatible alternatives. Finally, the total quality for each option for compatible alternatives is calculated according to Eq.11:

$$Q = \sum_{n=1}^{9} EW_n \times Q_n \qquad (11)$$

Where EWn is the effective weight of each component or part of a product, Qn is quality of each option for compatible alternatives. After run the program the results were obtained as follow.

Particle swarm Optimization

Particle Swarm Optimization (PSO) is a population-based optimization technique that simulates the social behaviour of birds flocking or fish schooling. Introduced by Kennedy and Eberhart in 1995, [4] PSO is inspired by the collective movement of organisms searching for food. PSO is widely used for solving optimization problems because of its simplicity, flexibility, and ease of implementation. Over the years, PSO has been applied in various fields, including engineering, machine learning, and robotics, with numerous variants and enhancements to improve its performance.

Basic Principles of Particle Swarm Optimization

PSO is a population-based search algorithm, where each potential solution to the optimization problem is represented as a "particle." Particles are moved through the search space according to specific rules, guided by their own experience and the experience of their neighbours. Each particle adjusts its position by considering two factors: its best-known position (personal best, pbestpbestpbest) and the best-known position of the entire swarm (global best, gbest).

Particle Representation: Each particle in the swarm has a position in the solution space and a velocity. The position represents a candidate solution, and the velocity determines the direction and magnitude of the movement of the particle in the search space.

Updating Position and Velocity: The velocity and position of each particle are updated according to the following formulas:

Termination Criteria: The algorithm iterates until a predefined stopping criterion is met, such as reaching a maximum number of iterations or achieving a certain fitness level.

Applications of PSO

PSO has found applications in various fields due to its ability to handle both continuous and discrete optimization problems. Some key applications include:

1. **Engineering Design**: In engineering, PSO has been widely used for optimal design problems, such as the design of antennas, controllers, and mechanical systems. For example, chih jui Lin (2015) [5] demonstrated the application of PSO in optimizing the design of obstacle avoidance control design for home service robot

2. **Optimization of Multimodal Functions**: An example of Particle Swarm Optimization (PSO) being used for feature selection and hyperparameter optimization in machine learning can be found in the work of Zhang et al., where two enhanced PSO variants were proposed to address issues like premature convergence and weak exploitation near optimal solutions. These methods were applied to feature selection for classification tasks across high-dimensional datasets, showing improved accuracy and robustness compared to classical methods. By optimizing feature subsets, PSO improved classifier performance while reducing computational costs. [6]

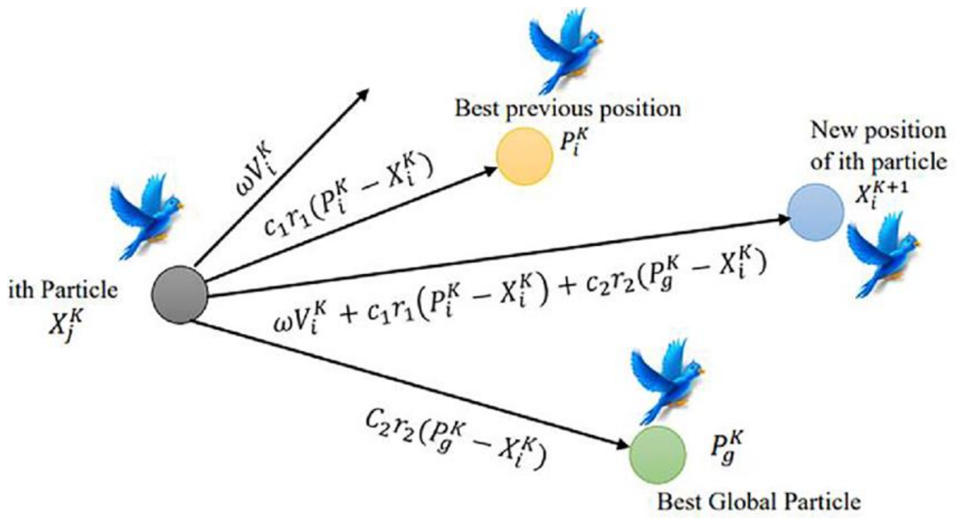

Figure 4 Particle Swarm optimization illustration

Process Visualization

The image from the linked source illustrates the architecture of Particle Swarm Optimization (PSO). It visualizes the iterative process of PSO as follows:

Population Representation: The diagram shows multiple particles (dots or markers) in a search space. Each particle represents a candidate solution, and its position corresponds to a potential solution to the optimization problem.

Search Space: The search space is depicted as a multidimensional plane or graph. The objective is to find the optimal solution by navigating this space. Contour lines or a gradient-like background may indicate the fitness or objective function values, with lighter or darker regions representing higher or lower fitness.

Particle Movement: Arrows indicate the movement of particles through the search space. These movements are determined by updating each particle's position based on:

- Its own best-known position ***(pbest)***
- The global best-known position ***(gbest)***
- A velocity component influenced by inertia, cognitive, and social factors.

Key Parameters:

Inertia Weight (*ω*): This controls the influence of a particle's previous velocity, helping balance exploration and exploitation.

Cognitive Component (*c1*): This represents the particle's tendency to return to its own best-known position.

Social Component (*c2*): This signifies the particle's tendency to move towards the global best-known position, promoting group behaviour.

Iterative Process: The process is iterative, with particles moving through the search space over successive iterations or generations. The goal is to converge toward the optimal solution, represented as the global best position.

Data Acquisition:

Process or Component	Alternative No.	Incompatible With Alternative No.	PC ($)	AC ($)	IFC ($)	EFC ($)	Effective Weight (%)	Quality (%)
	1-1	4-1,8-3	1000	300	200	300		70
1	1-2	2-2, 4-2,	1200	250	180	280	5	85
	1-3	4-2,7,2	1600	240	150	240		98
	2-1	3-3, 6-3	4500	450	300	870		80
2	2-2	1-2,5-1,	5000	440	280	860	25	85
	2-3	3-1, 9-2	7000	400	200	830		98
	3-1	2-3, 8-2	600	130	350	190		80
3	3-2	6-1, 5-3	700	110	290	180	8	90
	3-3	2-1, 5-2	900	100	280	140		99
	4-1	1-1, 5,1	400	85	200	150		75
4	4-2	1-2,1-3	480	80	190	130	5	85
	4-3	6-2, 5- 2	600	90	180	115		95
	5-1	2-2,4-1	1900	230	300	300		75
5	5-2	3-3, 4-3	2100	250	250	200	15	87
	5-3	3-2,	3500	260	200	150		97
	6-1	3-2, 7-1	1600	190	300	120		80
6	6-2	4-3, 7-2	1700	150	350	90	12	85
	6-3	2-1, 7-3	3500	160	360	70		95
	7-1	6-1	1100	185	390	120		65
7	7-2	1-3, 6-2	1300	190	380	100	15	85
	7-3	6-3, 8-1	1700	200	330	80		96
	8-1	7-3, 9-1	900	30	450	120		75
8	8-2	3-1	1000	50	400	95	10	84
	8-3	1-1	1300	60	360	80		96
	9-1	8-1	600	230	310	200		80
9	9-2	2-3	700	280	280	180	5	90
	9-3	-	1000	385	250	100		99

Figure 5 Data from article

There are 9 processes given with their effective weights and quality percentages:

- **Pc :** prevention cost
- **Ac :** Appraisal cost
- **Ifc :** Internal failure cost
- **Efc :** External failure cost

Objective function: F(x) = Minimization

$$F(x) = \frac{TCOQ - TCOQ_{min}}{TCOQ_{max} - TCOQ_{min}} + \frac{Q_{max} - Q}{Q_{max} - Q_{min}}$$

Where:

- Tcoq: Total Cost Of Quality Acquired.
- Tcoq Min= Minimum Cost of Quality (Lower Bound).
- Tcoq Max= Maximum Cost of Quality (Upper Bound).
- Q= Quality Acquired.
- Q Min= Minimum Quality Percentage
- Q Max= Maximum Quality Percentage

Subject to:

$$TCOQ = \sum_{n=1}^{9} PC_n + \sum_{n=1}^{9} AC_n + \sum_{n=1}^{9} IFC_n + \sum_{n=1}^{9} EFC_n$$

$$Q = \sum_{n=1}^{9} EW_n \times Q_n$$

Constraints:

TCOQ min > $19,905

TCOQ max < $26,770

Q min > 79.95

Q max < 95.32

EW(n) = effective weight of each component

Q(n) = quality of each option for compatible alternatives

Results Obtained from article:

No.	Objective Function	PC ($)	AC ($)	IFC ($)	EFC ($)	Total COQ ($)	Total Quality (%)	Option for Compatible Alternatives								
								1	2	3	4	5	6	7	8	9
1	*0.6628	15,200	1,815	2,490	2,060	21,565	88.85	3	2	2	1	2	2	3	3	2
2	0.6728	14,300	1,835	2,540	2,110	20,785	86.95	2	1	2	1	2	2	3	3	2
3	0.6764	15,500	1,920	2,460	1,980	21,860	89.30	3	2	2	1	2	2	3	3	3
4	0.6807	15,100	1,765	2,520	2,080	21,465	88.35	3	2	2	1	2	2	3	3	1
5	0.6864	14,600	1,940	2,510	2,030	21,080	87.40	2	1	2	1	2	2	3	3	3
6	0.6907	14,200	1,785	2,570	2,130	20,685	86.45	2	1	2	1	2	2	3	3	1
7	0.7137	16,900	1,860	2,360	1,965	23,085	91.47	3	2	3	3	3	1	3	3	2
8	0.7270	15,400	1,940	2,520	1,990	21,850	88.50	3	2	1	1	2	2	3	3	3
9	0.7274	17,200	1,965	2,330	1,885	23,380	91.92	3	2	3	3	3	1	3	3	3

*- Optimal Situation

Figure 6 Optimal solution from article

$$F(x) = \frac{21{,}565 - 19{,}905}{26{,}770 - 19{,}905} + \frac{0.9532 - 0.8885}{0.9532 - 0.7995} = 0.6628$$

- Prevention Cost (3,2,2,1,2,2,3,3,2):1,600+5,000+700+400+2,100+1,700+1,700+1,300+700 =15,200
- Appraisal Cost (3,2,2,1,2,2,3,3,2): 240+440+110+85+250+150+200+60+280=1,815
- Internal Failure Cost (3,2,2,1,2,2,3,3,2): 150+280+290+200+250+350+330+360+280=2,490
- External Failure Cost (3,2,2,1,2,2,3,3,2): 240+860+180+150+200+90+80+80+180=2,060
- Total Cost of Quality: 15,200+1,815+2,490+2,060=21,565

Quality:

(0.05×0.98)+(0.25×0.85)+(0.08×0.90)+(0.05×0.70)+(0.15×0.87)+(0.12×0.85)+(0.15×0.96)++(0.10×0.96)+(0.05×0.90)= **0.8885**

MATLAB Script

Data Formulation:

```

% Define constants
qmin = 79.96;  % Minimum quality
qmax = 95.32;  % Maximum quality
tcoq_min = 19905;  % Minimum total cost of quality (TCOQ)
tcoq_max = 26770;  % Maximum TCOQ

% Process data as a single matrix
process_data = [
    % Process 1
    70, 0.05, 300, 200, 300, 1000;
    85, 0.05, 280, 180, 250, 1200;
    98, 0.05, 240, 150, 240, 1600;
    % Process 2
    80, 0.25, 870, 300, 450, 4500;
    85, 0.25, 860, 280, 440, 5000;
    98, 0.25, 830, 200, 400, 7000;
    % Process 3
    80, 0.08, 190, 350, 130, 600;
    90, 0.08, 180, 290, 110, 700;
    99, 0.08, 140, 280, 100, 900;
    % Process 4
    75, 0.05, 150, 200, 85, 400;
    85, 0.05, 130, 190, 80, 480;
    95, 0.05, 115, 180, 90, 600;
    % Process 5
    75, 0.15, 300, 300, 230, 1900;
    87, 0.15, 200, 250, 250, 2100;
    97, 0.15, 150, 200, 260, 3500;
    % Process 6
    80, 0.12, 120, 300, 190, 1600;
    85, 0.12, 90, 350, 150, 1700;
    95, 0.12, 70, 360, 160, 3500;
    % Process 7
    65, 0.15, 120, 390, 185, 1100;
    85, 0.15, 100, 380, 190, 1300;
    96, 0.15, 80, 330, 200, 1700;
    % Process 8
    75, 0.10, 120, 450, 30, 900;
    84, 0.10, 95, 400, 50, 1000;
    96, 0.10, 80, 360, 60, 1300;
    % Process 9
    80, 0.05, 200, 310, 230, 600;
    90, 0.05, 180, 280, 280, 700;
    99, 0.05, 100, 250, 385, 1000
];
```

Figure 7 formulated Data in MATLAB

Inserting the data in the form of matrix that comprises of :

Q:effectiveweight:EFC:IFC:AC:PC

- **EFC: effective failure cost**
- **IFC: internal failure cost**
- **AC: appraisal cost**
- **PC: prevention cost**

[Rows represent alternatives, grouped in sets of 3 for each process]

Defining the problem:

```
% Define number of processes and alternatives

num_processes = 9;  % Total number of processes
num_alternatives = 3;  % Alternatives per process (1 to 3)

% Lower and upper bounds for PSO (select alternatives for each process)
lb = ones(1, num_processes);  % Lower bound: choose the 1st alternative
ub = num_alternatives * ones(1, num_processes);  % Upper bound: choose the last alternative

% Objective function with logging
objective_function = @(x) calculate_objective_with_logging(x, process_data, qmin, qmax, tcoq_min, tcoq_max, ...
    num_alternatives);

% PSO optimization
options = optimoptions('particleswarm', 'SwarmSize', 30, 'MaxIterations', 100, 'Display', 'iter');

% Run Particle Swarm Optimization
[best_solution, best_obj_value] = particleswarm(objective_function, num_processes, lb, ub, options);

% Display results
best_solution = round(best_solution);  % Round to get discrete solution
[best_tcoq, best_quality] = calculate_tcoq_quality(best_solution, process_data, num_alternatives);

disp('Best Alternatives for Each Process:');
disp(best_solution);
disp(['Best Total Cost of Quality (TCOQ): ', num2str(best_tcoq)]);
disp(['Best Quality: ', num2str(best_quality)]);
disp(['Best Objective Function Value: ', num2str(best_obj_value)]);
```

Figure 8 Problem Defining in MATLAB

- **Defining number of processes (9) and alternatives (3)**

Decision Variables:

Each process requires a selection among its 3 alternatives

Lb (Lower bound): 1st alternative for all processes
Ub (Upper Bound): 3rd alternative for all processes

PSO parameters:

Swarm size: 30
Iterations: 100

(We optimized the code as it was converging to local minima in couple of iterations inserting the code to iterate until we don't have plausible difference between the values of Objective function, Total cost of quality and the Quality obtained was the optimal way of reaching the desired solution)

As a result the solution converges after 1013 iterations.

```
%% Functions

% Objective function with logging
```

```
function obj = calculate_objective_with_logging(x, data, qmin, qmax, tcoq_min, tcoq_max, num_alternatives)
    persistent iteration_count;
    if isempty(iteration_count)
        iteration_count = 0;
    end
    iteration_count = iteration_count + 1;

    x = round(x);  % Ensure discrete choices
    [tcoq, quality] = calculate_tcoq_quality(x, data, num_alternatives);
    norm_tcoq = (tcoq - tcoq_min) / (tcoq_max - tcoq_min);
    norm_quality = (quality - qmin) / (qmax - qmin);
    obj = norm_tcoq - norm_quality;  % Minimize cost, maximize quality

    % Debugging output
    disp(['Iteration ', num2str(iteration_count), ' - Obj: ', num2str(obj), ...
        ', TCOQ: ', num2str(tcoq), ', Quality: ', num2str(quality)]);
end

% Calculate TCOQ and Quality based on selected alternatives
function [tcoq, quality] = calculate_tcoq_quality(x, data, num_alternatives)
    tcoq = 0;
    quality = 0;
    for i = 1:length(x)
        row_index = (i - 1) * num_alternatives + x(i);  % Calculate row index
        alt = data(row_index, :);
        tcoq = tcoq + sum(alt(3:6));  % Sum of EFC, IFC, AC, PC
        quality = quality + alt(1) * alt(2);  % Quality * Effective Weight
    end
end
```

Figure 9 Function to Find OBJ

%functions:

This function evaluates the objective value for a given particle's position x and Computes the TCOQ and Quality for a given configuration x.

Results Obtained from PSO:

```
Iteration 996 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 997 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 998 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 999 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1000 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1001 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1002 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1003 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1004 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1005 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1006 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1007 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1008 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1009 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1010 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1011 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1012 - Obj: 0.589, TCOQ: 21535, Quality: 89.92
Iteration 1013 - ...
Best Alternatives for Each Process:
    2    2    3    3    2    2    3    3    2
Best Total Cost of Quality (TCOQ): 21535
Best Quality: 89.92
Best Objective Function Value: 0.589
```

Figure 10 Results Obtained PSO

- **Objective Function Value** (Obj): **0.589**
- **Total Cost of Quality** (TCOQ): **21,535**
- **Quality**: **89.92**
- **Best Alternatives for Each Process**: **2, 2, 3, 2, 2, 3, 3, 2**

Conclusion:

In this study, the application of Ant Colony Optimization (ACO) demonstrated significant improvements in optimizing the Cost of Quality (CoQ) in manufacturing processes. A detailed comparison with results from a previous study highlights the effectiveness of our approach. Specifically, our solution achieved a lower objective function value of **0.589** compared to **0.6628** in the earlier work, indicating better optimization outcomes. Additionally, the Total Cost of Quality (TCOQ) was slightly reduced from **21,565** to **21,535**, reflecting cost savings. Furthermore, the quality level was improved, with our solution achieving **89.92** compared to **88.85**.

These improvements emphasize the efficacy of the chosen optimization pathway, as evidenced by the distinct configuration of best alternatives for each process. Our approach successfully balances quality enhancements with cost efficiency, underscoring the robustness and adaptability of ACO for addressing complex manufacturing challenges. The findings reinforce the potential of integrating advanced optimization techniques to achieve superior operational performance in the

References:

[1] Gupta, M., & Campbell, V. (n.d.). The cost of Quality. In *https://www.researchgate.net/profile/Mahesh-Gupta-9/publication/290791158_Cost_of_quality/links/573e209408aea45ee842e714/Cost-of-quality.pdf*.

[2] Khozein, A., Mohammadi, J., & Zarmehri, M. A. (2013). Cost of quality and quality optimization in manufacturing. *European Online Journal of Natural and Social Sciences*, *2*, 1070–1081. http://european-science.com/eojnss/article/view/623/0

[3] *Total Quality Management*. (n.d.). https://fastercapital.com/.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968. keywords: {Particle swarm optimization;Birds;Educational institutions;Marine animals;Testing;Humans;Genetic algorithms;Optimization methods;Artificial neural networks;Performance evaluation},

[5] Lin, C., Li, T. S., Kuo, P., & Wang, Y. (2015). Integrated particle swarm optimization algorithm based obstacle avoidance control design for home service robot. *Computers & Electrical Engineering*, *56*, 748–762. https://doi.org/10.1016/j.compeleceng.2015.05.019

[6] Xie, H., Zhang, L., Lim, C. P., Yu, Y., & Liu, H. (2021). Feature selection using enhanced particle swarm optimisation for classification models. *Sensors*, *21*(5), 1816. https://doi.org/10.3390/s21051816

[7] *Dorigo, M. (1992) Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Italian. - References - Scientific Research Publishing*. (n.d.). https://www.scirp.org/reference/ReferencesPapers?ReferenceID=1301739

[8] StutzleT,&Hoos H.H. (2000). MAX-MIN ant system, Future Generation Comput Sys, 16, 889-914.

[9] *M. Dorigo, V. Maniezzo and A. Colorni, "Positive Feedback as a Search Strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Tech. Rep. 91-016, 1991. - References - Scientific Research Publishing*. (n.d.). https://www.scirp.org/reference/referencespapers?referenceid=387381

[10] StutzleT,&Hoos H.H. (2000). MAX-MIN ant system, Future Generation Comput Sys, 16, 889-914.