P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia
Science & Technology Asia
Vol.29 No.4 October - December 2024
Page: [184-196]
Original research article

Implementation of Particle Swarm Algorithm to Sine-Cosine Optimization: Case Study of Plant-Wide Chemical Processes

Orn-anong Winyutrakoon, Panida Chunhakamolrak, Thongchai Rohitatisha Srinophakun*

Department of Chemical Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand

Received 6 September 2024; Received in revised form 13 November 2024
Accepted 19 November 2024; Available online 27 December 2024

ABSTRACT

The sine-cosine algorithm (SCA) is a population-based optimization approach that relies on mathematical models incorporating sine and cosine functions to effectively obtain the global optimum. The SCA necessitates multiple initial solutions to fluctuate outward and toward the best solution. In this study, the particle swarm algorithm was replaced with randomization and selection of the best fit to enhance SCA operations, resulting in faster convergence and improved accuracy, thereby termed the modified SCA (MSCA). The proposed algorithm's performance was investigated and compared using 7 general problems, 19 common benchmarks, and 5 industrial-scale chemical processes. These benchmarks varied in complexity, with and without constraints. For comparative purposes regarding the algorithm's effectiveness, the results were averaged based on a maximum of 1,000 iterations, utilizing 500 populations from 5 replications. Subsequently, 5 common chemical processes were used to demonstrate the implementation using commercial simulation software (Aspen Plus), seamlessly integrated with both optimization algorithms to adjust operational conditions and design parameters. MSCA produced a lower error percentage and CPU time than SCA.

Keywords: Benchmark; Heuristic; Metaheuristic; Particle Swarm Optimization (PSO); Sine Cosine Algorithm (SCA)

1. Introduction

Optimization involves applying specific methods to determine the best (including most cost-effective) solution to a problem or to aid in process design. Optimization can address a wide array of challenges in chemical plant design, construction, operation, and analysis. This

***Corresponding author:** fengtcs@ku.ac.th

technique has found extensive application across various fields, including science, business, and, notably, engineering [1]. There are two primary categories: conventional methods (typically involving calculus) and non-conventional methods. Calculus-based methods are well-suited for simple discrete problems, while non-conventional methods, though limited in practice, are devised to offer faster problem-solving. As problems become more complex and non-linear, additional constraints can be applied when the calculus method fails to find an exact solution, albeit with added complexity. Non-conventional methods can be further categorized into heuristic and modified heuristic techniques. The heuristic approach involves problem-specific strategies to approximate solutions, often applying single randomizations for local optimization. On the other hand, a metaheuristic method is a problem-independent technique that approximates global optimization through multiple randomizations [2-5].

The current study focused on a metaheuristic algorithm utilizing mathematical models and equations to design an optimization algorithm. We specifically chose the sine cosine algorithm (SCA) and another approach known as particle swarm optimization (PSO), the latter modeling swarming behavior, to propose optimization algorithms. These two metaheuristic algorithms were implemented through code developed in the MATLAB software package, a methodology previously applied to solve benchmark and chemical engineering problems.

2. Methodology

2.1 Sine and cosine algorithm

The sine and cosine functions serve as the primary equations in the process, guiding the steps of exploration and exploitation. The solution space is defined as the area likely to contain the optimum solution, with the most effective search and exploitation process involving continuous adjustment of calculated results within the solution space to identify the most suitable answer. The SCA process applies position adjustments based on Eq. (2.1):

$$X_i^{t+1}=\begin{cases} X_i^t+r_1\times\cos(r_2)\times\left|r_3P_i^t-X_i^t\right| \;;\; r_4<0.5 \\ X_i^t+r_1\times\sin(r_2)\times\left|r_3P_i^t-X_i^t\right| \;;\; r_4\geq 0.5 \end{cases}, \tag{2.1}$$

where X_i^t is the position of the current solution in the i^{th} dimension at the t^{th} iteration, P_i^t is the position of the destination (the best solution obtained from a past iteration), and r_i is the parameter.

The SCA process incorporates four main parameters: r_1, r_2, r_3, and r_4. Parameter r_1 determines the region or movement direction for the next position, whether it lies between the solution and the destination or outside it, while r_2 specifies the distance of the movement toward or away from the destination. The parameter r_3 introduces a random weight for the destination, stochastically emphasizing ($r_3 > 1$) or de-emphasizing ($r_3 < 1$) its effect by defining the distance. Lastly, r_4 equally toggles between sine and cosine functions.

Operation of the SCA model commenced with the randomization of initial values using an initializing function. Subsequently, initial values for the problem defined in the objective function were calculated and stored to track value changes. Following adjustment according to the SCA function, iterative cycles ensued until the maximum was reached [5–6].

2.2 Particle swarm optimization

Swarm optimization is a stochastic, population-based search method that mimics the behavior of fish schooling, birds flocking, and other grouping behaviors [7]. Often, a PSO algorithm is developed by modifying the variables in the PSO equation by changing the constants to complex functions.

2.3 Modified sine cosine algorithm

In testing the sine cosine algorithm's calculations, parameters are continuously randomized, causing the calculations to take a long time and slowing down the search for suitable solutions. Therefore, we introduced another algorithm to help reduce the randomization of new parameters by selecting appropriate values before accepting the randomization of new parameters of the SCA. When choosing another algorithm, we opted for PSO, which has efficient calculations and effectiveness in finding solutions. The SCA has parameters that are difficult to tune properly, with poor configuration reflecting weak exploitation. However, these parameters do not affect the exploration of the search space. The structure of the proposed approach considers layers, as shown in Fig. 1, where the top layer contains y particles, with their movement performed by the PSO operators. This step proposes the suitable population, Y^M, and M values. Each of them goes through the comparison with population, X^{YN}, and N values by the SCA algorithm in the bottom layers. The bottom layer separates the population into y groups, each containing N search agents; the new positions are computed using the SCA. The best solution found by each group is kept by a particle in the top layer. Hence, the bottom layer focuses on exploring the search space, while the top layer focuses on exploiting the best solutions found by the bottom layer. This hybridization scheme belongs to the classification of high-level and co-evolutionary hybrid meta-heuristic models [8].

As shown in Fig. 2, the MSCA flowchart begins by initializing the required parameters of the SCA and PSO functions and randomizing the initializing search agent (x_{ij}) using the 'Initialize' function. Then, the objective function is calculated, x values are stored and converted to Y_i before finding the minimum fitness value by evaluating particles as Y_{gBest}. Subsequently, the process iteratively loops to evaluate the new fitness value using the SCA function. If ($x_{ij} < Y_i$), then $Y_i = x_{ij}$ will update with new values for r_1, r_2, r_3, and r_4. However, if not, the latest r_1, r_2, r_3, and r_4 values are used. The new x_{ij} values are then used to find the Y_i value by evaluating the fitness value using the PSO function. These new Y_i values can update the Y_{gBest} value. The maximum iteration will approach the final objective value. The codes with examples are available at https://drive.google.com/drive/folders/1dug2r07ER2IUD0U9AwARMH4xjiAokV_6?usp=sharing.

SCA parameters(r_i) still follow their formulas, while PSO introduces the best-nominated population (the particular set that gives the best fit) for SCA. Simply said, before running through the SCA, we proposed the best input using PSO. Then, the process passes through SCA steps and returns to PSO to find the best-fit solution. This modified step can enhance the optimum search for the whole process, as detailed in Fig. 2.

3. Results and Discussion

3.1 Standard benchmarks

The modified SCA was validated using 7 benchmark problems [9]. Table 1 provides additional applications to 5 MATLAB and Aspen Plus problems, as well as the process plant layout. The benchmark and 5 chemical engineering problems were classified as multivariable and nonlinear, with or without constraints [10]. This section presents the supplied data for each benchmark and the results from applying the SCA and MSCA codes. Each problem was repeated 5 times; the mean values of the answers are provided in Table 2.

An additional 19 benchmarks with codes and comparisons are provided in the share drive. This proposed algorithm was implemented seamlessly with the industrial process simulation. Five standard processes, from basic to plant-wide chemical processes, are covered. We perform these 1,000 iterations from the beginning of the SCA and shape up by checking the fitness by PSO. By the way, we used this 1,000 iterations criterion on the benchmarks to guarantee our performance of MSCA.

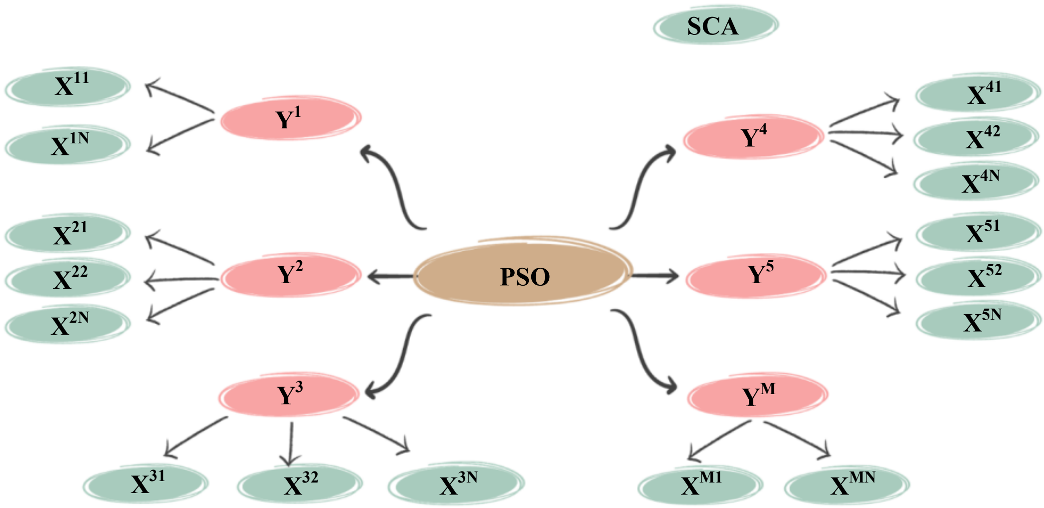


Fig. 1. Structure of proposed adaptive MSCA.

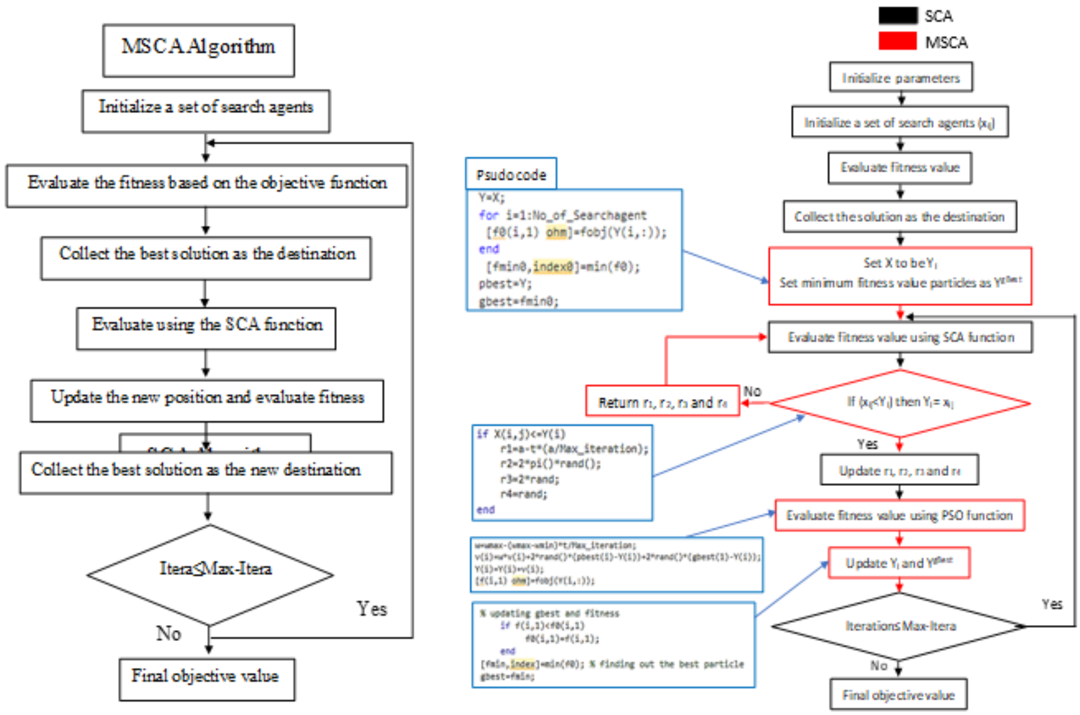
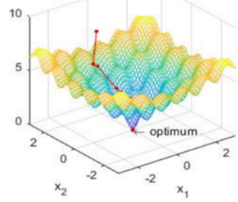
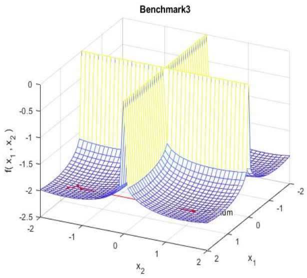
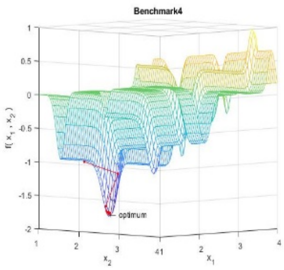
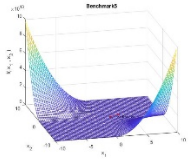
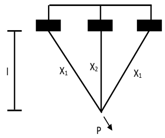
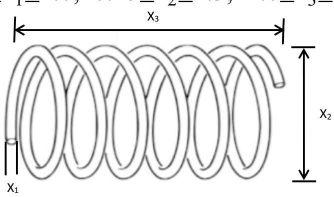


Fig. 2. SCA and MSCA flowcharts.

Table 1. Objective functions of 7 benchmarks.

No.	Name	Objective function
		No constraint problems
B1	Ackley d=2 and d=5	Minimize function: $f(x) = -20\exp\left((-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$ 
B2	Cross-in-Tray	Minimize function: $f(x) = -0.0001\left(\left\|\sin(x_1)\sin(x_2)\exp\left(\left\|100-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\|\right)\right\| + 1\right)^{0.1}$ 
B3	Michalewicz d=2, d=5, and d=10	Minimize function: $f(x) = -\sum_{i=1}^{d}\sin(x_i)\sin^{20}\left(\frac{ix_i^2}{\pi}\right)$ 
		Constraint problems

B4	Keane's bump m=2 m=5 m=20	Minimize function: $f(x) = -\left\|\left\{\sum_{i=1}^{d}\cos^4(x_i) - 2\prod_{i=1}^{d}\cos^2(x_i)\right\}/\left(\sum_{i=1}^{d} ix_i^2\right)^{0.5}\right\|$ $g_1(x){:}0.75-\prod_{i=1}^{m}x_i<0,\ g_2(x){:}\ \sum_{i=1}^{m}x_i-7.5m<0$ $0 < x_i < 10$ 
B5	Three-bar truss design	Minimize function: $f(x) = \left(2\sqrt{2}x_1 + x_2\right)100$ x_1, x_2 = Area of the truss (cm^2) $g_1(x) = \frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma \leq 0$ $g_2(x) = \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma \leq 0$ $g_3(x) = \frac{1}{\sqrt{2}x_2+x_1}P-\sigma \leq 0$ 
B6	Compressional and tensional spring	Minimize function: $f(x) = x_1^2x_2(2+x_3)$ x_1 = Wire diameter (inches) x_2 = Coil diameter (inches) x_3 = Coil length (inches) $g_1(x)=1-\frac{x_2^3x_3}{71785x_1^4} \leq 0$ $g_2(x)=\frac{4x_2^2-x_1x_2}{12566(x_1^3x_2-x_1^4)}+\frac{1}{5108x_1^2}-1 \leq 0$ $g_3(x) = 1-\frac{140.45x_1}{x_2^3x_3} \leq 0$ $g_4(x) = \frac{x_1+x_2}{1.5}-1 \leq 0$ $0.05 \leq x_1 \leq 2.0,\ 0.25 \leq x_2 \leq 1.3,\ 2.0 \leq x_3 \leq 15.0$ 

B7 Welded beam design

Minimize function: $f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$

x_1 = Weld height (inches)

x_2 = Weld length (inches)

x_3 = Beam height (inches)

x_4 = Beam length (inches)

$$g_1(x) = \tau(x) - \tau_{max} \leq 0$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0$$

$$g_3(x) = x_1 - x_4 \leq 0$$

$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0$$

$$g_5(x) = 0.125 - x_1 \leq 0$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0$$

$$g_7(x) = P - P_c(x) \leq 0$$

$$\tau(x)=\sqrt{(\tau')^2+\tau'\tau''\frac{x_2}{2R}+(\tau'')^2}$$

$$\tau'=\frac{P}{\sqrt{2}x_1x_2},\quad \tau''=\frac{MR}{J},\qquad M=P\left(L+\frac{x_2}{2}\right)$$

$$R=\sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},\qquad \delta(x)=\frac{4PL^3}{Ex_3^3x_4}$$

$$J=2\left[\sqrt{2}x_1x_2\left\{\frac{x_2^2}{12}+\left(\frac{x_1x_3}{2}\right)^2\right\}\right],\quad \sigma(x)=\frac{4PL}{x_3^2x_4}$$

$$P_c(x)=\frac{4.013\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$P = 6000$ lb, $L = 14$ in, $E = 30 \times 10^6$ psi

$G = 12\times10^6$ psi, $\tau_{max} = 13{,}600$ psi, $\sigma_{max} = 30{,}000$ psi

$\delta_{max} = 0.25$ in, $0.1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$

$0.1 \leq x_3 \leq 10$, $0.1 \leq x_4 \leq 2$

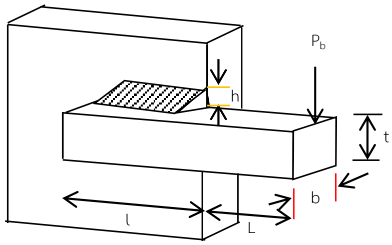


Table 2. Objective functions of 7 benchmarks.

Benchmark	Variable	Solution	SCA			MSCA		
			CPU time (s)	Average result	Error (%)	CPU time (s)	Average result	Error (%)
B1	2	0.0000	19.4715	8.8818e-16	0.0000	2.3567	8.8818e-16	0.0000
	5	0.0000	26.1469	8.8818e-16	0.0000	2.5254	8.8818e-16	0.0000
B2	2	-2.0626	20.0556	-2.0626	0.0000	1.8528	-2.0626	0.0000
B3	2	-1.8013	50.7376	-1.8011	0.0100	5.2119	-1.8013	0.0000
	5	-4.6877	62.9467	-3.1858	32.0369	5.3740	-4.2122	10.1416
	10	-9.6602	84.7882	-5.3232	44.8953	5.3740	-7.3012	24.4194
B4	2	-0.3650	67.0946	-0.3646	0.1041	5.5828	-0.3649	0.0274
	5	-0.6344	83.2254	-0.6176	2.6513	6.1538	-0.6327	0.2680
	20	-0.7600	149.0561	-0.4125	48.6710	9.3783	-0.7075	11.9587
B5	2	263.8958	54.0650	263.8561	0.0150	4.8679	263.8538	0.0159
B6	3	0.0127	71.0513	0.128	0.5020	5.9037	0.128	0.5020
B7	4	1.7249	80.8015	1.7751	2.9103	9.9841	1.7410	0.9334

3.2 Ethanol column optimization problem

Optimization of the ethanol column concerning capital expenditure and operating expenditure was performed using the Aspen Plus and MATLAB software packages. Three optimal operations (Extracted Stages, Reflux Ratio, and Feed Stage data) were utilized to calculate both capital expenditure (CAPEX) and operating expenses (OPEX). CAPEX represents the primary purchases a company makes for long-term use, while OPEX denotes the day-to-day expenses incurred to maintain business operations. The enhancement to SCA incorporated the ethanol column data. The proposed algorithm interacted with steady data from simulations and refined specific conditions to approach optimality.

Those questions were candidates by SCA and MSCA, then simultaneously communicated to Aspen Plus to calculate ethanol purity, ethanol recovery, OPEX, and CAPEX values at the distillation tower (as illustrated in Fig. 3). The values were adjusted iteratively until reaching the optimum objectives, as depicted in Fig. 4, with the optimum configurations of SCA (at a number of stages=13, reflux ratio=1.82, feed stage location=8) and MSCA (at a number of stages=15, reflux ratio=1.01, feed stage location=7). Notice that both simulations assume to be at optimum values after 8 iterations, and the MSCA cannot get better CAPEX.

3.3 Heat exchanger network

This objective was to determine the minimum area of three heat exchangers, as depicted in Fig. 5. These three units were responsible for heating a cold stream with a heat capacity flow rate of 100,000 kW/°F from 100 to 500 °F. The overall coefficients of heat transfer for heat exchangers 1, 2, and 3 were 120, 80, and 40 kW/ft²°F, respectively. MSCA was utilized to optimize the area of each heat exchanger, aiming for a minimum total area.

The objective function depicted in Fig. 5 provided the areas of heat exchangers 1 and 2. Then, these values were sent to Aspen to calculate the area of heat exchanger 3 and the temperature output from heat exchangers 1 and 2. This enabled a comparison between the calculated values and the set temperature when using the SCA and MSCA functions.

SCA and MSCA were utilized to determine the minimum area of each of the three heat exchangers, with calculations based on the same decision variables in each iteration. These decision variables were the areas of heat exchangers 1, 2, and 3, as well as the outlet temperatures of heat exchangers 1 and 2. Table 3 concluded the design values at these operations; additional minimum areas of the heat exchangers obtained from SCA and MSCA were 7,049.30 and 7,049.25 ft², respectively. Even though there is no significant difference in those values, MSCA performed slightly better than SCA in terms of computation time.

3.4 Bioethanol production from corn

Bioethanol stands out as one of the most promising biofuels derived from renewable resources. Among various biomass materials suitable for bioethanol production, corn has been chosen as the preferred feedstock for a bioethanol production plant due to its status as the most abundantly produced agricultural crop, as depicted in Fig. 6. The results indicated the plant could produce 0.63 million barrels of ethanol per year using 249,278 lb/hr. of corn as input.

Based on Fig. 6, the boil-up ratio of rectification, the reflux ratio of rectification, and the fermentation temperature were selected by SCA and MSCA algorithms and then sent to Aspen to calculate the maximum purity of bioethanol shown in Table 4.

The bioethanol production from the corn problem was unconstrained, with the production calculated using the same decision variables in each iteration, namely the boil-up ratio of rectification, the reflux ratio of rectification, and the fermentation temperature. The maximum purity percentages of bioethanol obtained using the SCA and MSCA approaches were 98.78 and 99.14%, respectively, with approximately half the CPU time.

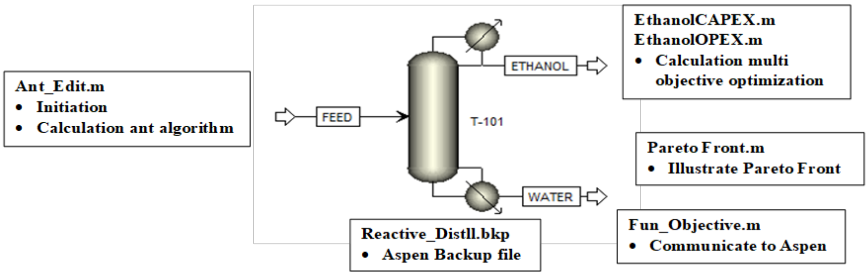


Fig. 3. Optimization of ethanol column.

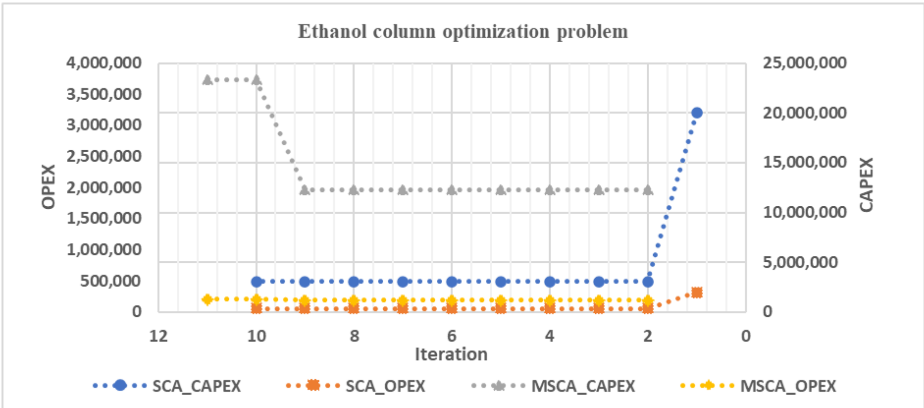


Fig. 4. Results from SCA and MSCA applications to ethanol column optimization problem.

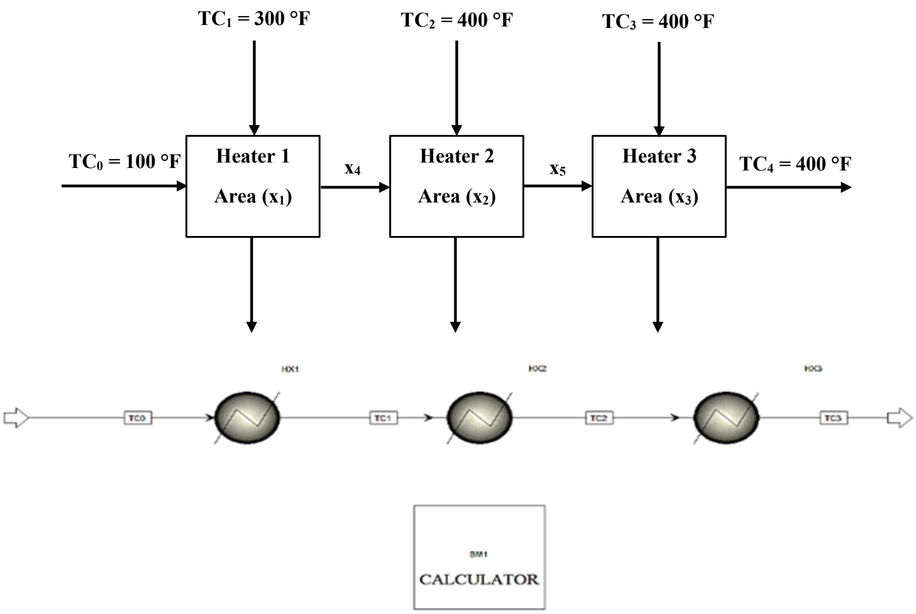


Fig. 5. Heat exchanger network design.

Table 3. Results for heat exchanger networks problem using SCA and MSCA.

Result of chemical engineering I	SCA	MSCA
Minimum area of three heat exchangers; ft^2	7,049.30	7,049.25
Area of heat exchanger 1; ft^2	584.42	576.46
Area of heat exchanger 2; ft^2	1,358.88	1,360.79
Area of heat exchanger 3; ft^2	5,106.00	5,112.00
Temperature outlet of heat exchanger 1; °F	182.44	181.78
Temperature outlet of heat exchanger 2; °F	295.76	295.52
CPU Time; sec	26.082	6.8747

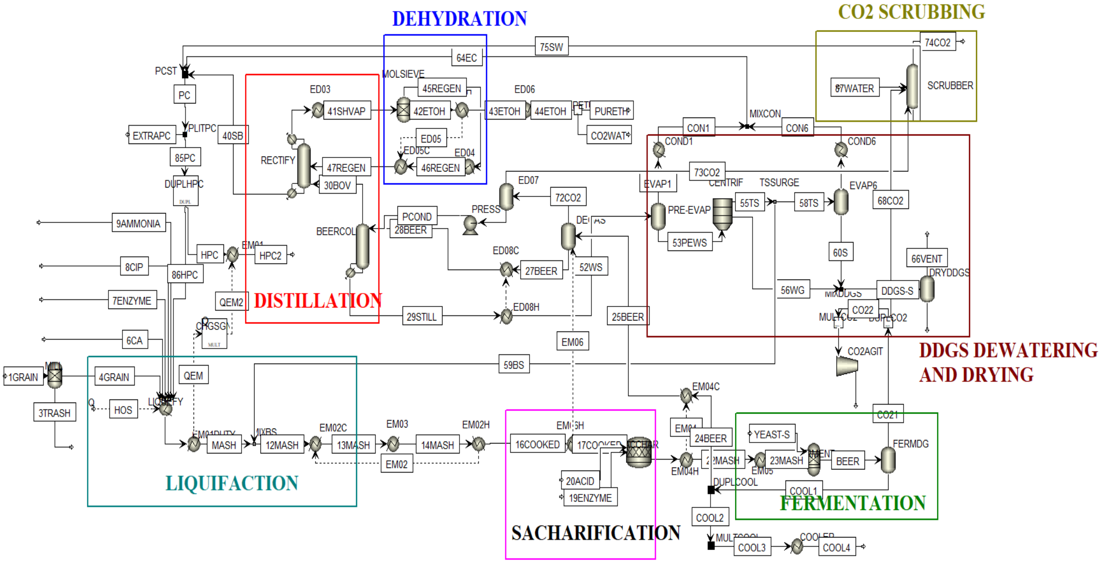


Fig. 6. Flow diagram of bioethanol production from corn.

Table 4. Results for bioethanol production using SCA and MSCA.

Result	SCA	MSCA
Maximum purity of bioethanol; %	98.78	99.14
Boil-up ratio of rectification	1.03	1.27
Reflex ratio of rectification	2.72	3.73
Fermentation temperature; °F	17.87	18.67
CPU time; sec	47.58	26.71

3.5 Biodiesel production from vegetable oil

Biodiesel, a clean-burning fuel, is produced through the transesterification of oils to create fatty acid methyl ester (FAME). Among the various methods used for FAME synthesis, the most common is alkali-catalyzed transesterification of vegetable oils and fats. Numerous types of feedstock oils can be utilized for biodiesel synthesis. In the case study process, the quantities of raw materials used were ethanol at 127.117 kg/hr., 10% NaOH at 50 kg/hr., oil at 1,050 kg/hr, and water at 50 kg/hr. The biodiesel production outcome depends on the FAME, as shown in Fig. 7.

The objective function maximized biodiesel yield by optimizing the reactor temperature, NaOH concentration, and residence time, as illustrated in Fig. 7. The constraints included ensuring that the concentration of FAME was greater than 96.5% wt., monoglycerides were less than 0.8% wt., and diglycerides and triglycerides were less than 0.2% wt. Additionally, the FAME density at 15°C had to be within the

860–900 kg/m³ range. These values were computed using the SCA and MSCA methods, and the resulting values are shown in Table 5.

The biodiesel production was calculated using the same decision variables in each iteration. The calculation of biodiesel production utilized consistent decision variables in each iteration. The maximum biodiesel yields based on SCA and MSCA were 98.75% wt. and 99.86% wt. It can be concluded that MSCA gives a slightly better yield than SCA. As with the trend in previous cases, MSCA required shorter computational time.

3.6 Safety process plant layout of empty-fruit-bunch biodiesel production

An empty-fruit-bunch biodiesel production plant can be divided into 7 sections: pretreatment, combustion, pyrolysis, separation, hydrotreating, distillation, and hydrocracking. Each section is operated under the same constraints, which must take into account the floor area of each piece of equipment and safety aspects based on the Dow Fire and Explosion Index system. The data inputs for this step are shown in Fig. 8. The mathematical model detailing the individual plant layout was expressed through the equations using a MILP approach. This approach aimed to determine both the safe process plant layout and the total cost, including connection cost, land cost, actual probable property damage cost, and protection device cost [11–12].

SCA and MSCA were applied to determine the optimal safety process plant layout for empty-fruit-bunch biodiesel production to identify the minimum total cost using the same decision variables in each iteration. The decision variables included equipment items' relative distances and total rectilinear distances across the 7 sections. Numerous constraints were imposed on this problem, considering the plan area of each equipment item and safety aspects utilizing the Dow Fire and Explosion Index system. As shown in Table 6, the minimum total costs based on SCA and MSCA were the same. However, the CPU time for MSCA was better (lower) than SCA's.

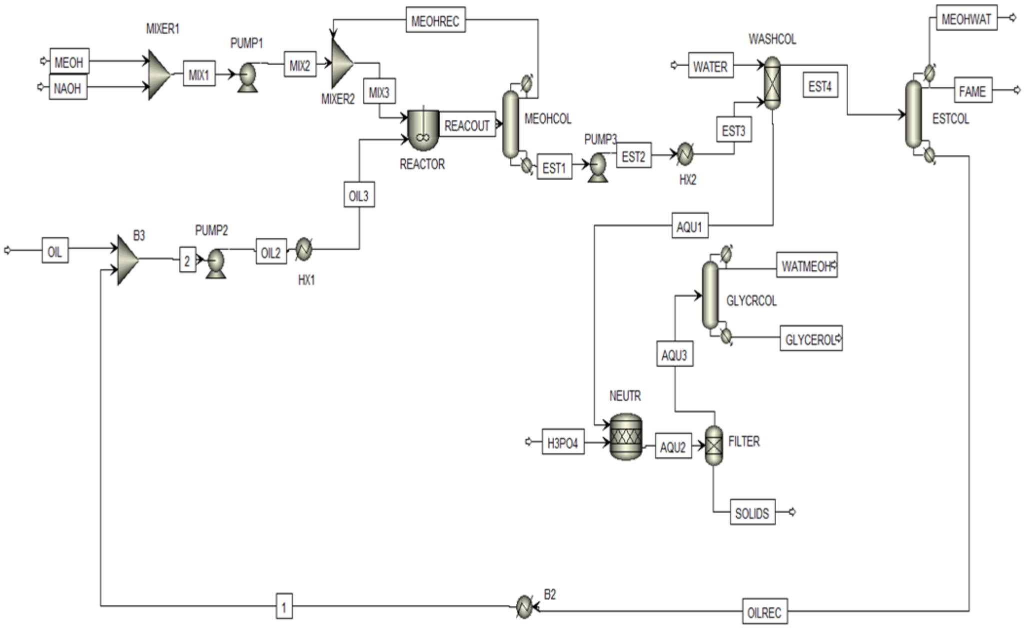


Fig. 7. Biodiesel production from vegetable oil.

Table 5. Result for biodiesel production using SCA and MSCA.

Result	SCA	MSCA
Maximum yield of biodiesel; %wt	98.75	99.86
Reactor temperature; °C	56.78	64.19
NaOH concentration; %wt	14.93	3.73
Residence time; hr	1.36	1.41
CPU time; sec	37.15	25.24

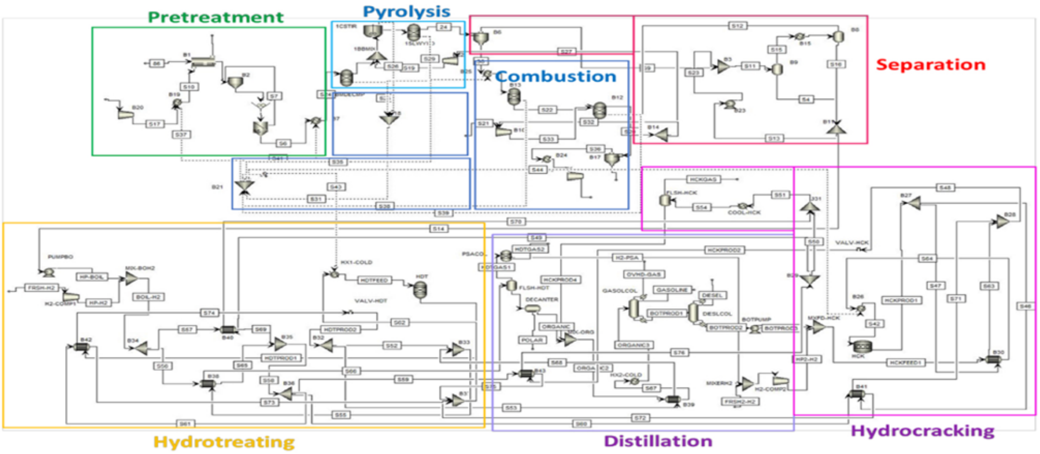


Fig. 8. Plant layout for biodiesel production from empty fruit bunches.

Table 6. Results for biodiesel production using SCA and MSCA.

Section	SCA		MSCA	
	Cost ($)	CPU Time (s)	Cost ($)	CPU Time (s)
1	1,299,106.02	0.684	1,299,106.02	0.220
2	3,555,729.23	0.581	3,555,729.23	0.195
3	5,523,333.51	0.497	5,523,333.51	0.210
4	10,652,155.05	0.456	10,652,155.05	0.201
5	635,361.40	0.472	635,361.40	0.290
6	13,894,311.58	0.505	13,894,311.58	0.199
7	505,721.09	0.428	505,721.09	0.208

4. Conclusion

The case studies demonstrated how the conventional SCA population and parameters can be enhanced using the PSO algorithm, known as MSCA. Thus, the best solution from each group (set of agents) generated by PSO was considered and then returned as input to the SCA algorithm. MSCA achieved faster convergence to an optimum by reducing parameter randomization. In conclusion, the various benchmarks were used based on the mean values obtained from 5 repetitions. MSCA produced lower error percentages and CPU times that could be attributed to selecting the best value population during the selection steps based on the particle swarm algorithm for 7 standard benchmarks. Additionally, there were 5 plant-wide processes linked to commercial simulation software, varying from simple unit operations to plant-wide processing. This concept could be further implemented in complex processes, such as analyzing techno-economic problems for sustainable development.

Declarations Author Contribution

O.W. and P.C. created the code, ran the experiment, prepared the figures, wrote the main manuscript, and T.S. reviewed the manuscript.

Availability of Material Data and Codes

https://drive.google.com/drive/u/2/folders/1dug2r07ER2IUD0U9AwARMH4xjiAokV6?%20usp=sharing

References

[1] Eduardo DG, Michael TK, Thomas FE. Optimization of chemical processes. 2nd ed. McGraw-Hill; 2001.

[2] Manish S, Dhaka V, Poonia RC. Metaheuristics for Communication Protocols: Overview and Conceptual Comparison. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2014;353-360.

[3] Winyutrakoon O, Rattanapairom S, Petpraphan P, Srinophakun TR. Implementation of Covariance Function to Improve Ant Colony Algorithm for Common Chemical Engineering Optimization. Eng. Sci., 2024:31(1140); 1-9.

[4] Sukpancharoen S, Srinophakun TR, Aungkulanon P. Grey Wolf Optimizer (GWO) with Multi-Objective Optimization for Biodiesel Production from Waste Cooking Oil Using Central Composite Design (CCD). Int. J. Mech. Eng. Robot. Res. 2020:9(8);1219-25.

[5] Seyedali M. SCA: A Sine Cosine Algorithm for solving optimization problems. Knowl.-Based Syst. 2016:96;120-33.

[6] Seyedali M. SCA: A Sine Cosine Algorithm. https://www.mathworks.com/matlabcentral/fileexchange/54948-sca-a-sine-cosine-algorithm. 2023.

[7] James K, Russell E. Particle Swarm Optimization. https://www.cs.tufts.edu/comp/150GA/homeworks/hw3/reading6 %201995%20particle%20swarming. 1995.

[8] Mohamed I, Aboul EH, Diego O, et al. ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. Expert Syst. Appl. 2018;99:56-70.

[9] Jamil M, Xin-She Y. A literature survey of benchmark functions for global optimization problems. Int. J. Math. Model. Numer. Optim. 2013:4(2);150-94.

[10] Simge E, Pakize E, Busra O. Solving Constrained Optimization Problems with Sine-Cosine Algorithm. Period. Eng. Nat. Sci. 2017:5(3);378-86.

[11] Papageorgiou LG, Rotstein GE, Continuous-Domain Mathematical Models for Optimal Process Plant Layout. Indust. Eng. Chem. Res. 1998:37(9);3631-9.

[12] Patsiatzis DI, Papageorgiou LG. Optimal multi-floor process plant layout. Comput. Chem. Eng. 2002:26(4);575-83.