

## Article

# Thermal-Economic Optimization of Plate–Fin Heat Exchanger Using Improved Gaussian Quantum-Behaved Particle Swarm Algorithm

Joo Hyun Moon <sup>1</sup>, Kyun Ho Lee <sup>1,2,\*</sup>, Haedong Kim <sup>1</sup> and Dong In Han <sup>3</sup>

<sup>1</sup> Department of Aerospace Engineering, Sejong University, Seoul 05006, Korea; jhmoon9@sejong.ac.kr (J.H.M.); haedong@sejong.ac.kr (H.K.)

<sup>2</sup> Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Korea

<sup>3</sup> Korea Aerospace Research Institute, Daejeon 34133, Korea; dihan@kari.re.kr

\* Correspondence: khlee0406@sejong.ac.kr

**Abstract:** Heat exchangers are usually designed using a sophisticated process of trial-and-error to find proper values of unknown parameters which satisfy given requirements. Recently, the design of heat exchangers using evolutionary optimization algorithms has received attention. The major aim of the present study is to propose an improved Gaussian quantum-behaved particle swarm optimization (GQPSO) algorithm for enhanced optimization performance and its verification through application to a multivariable thermal-economic optimization problem of a crossflow plate–fin heat exchanger (PFHE). Three single objective functions: the number of entropy generation units (*NEGUs*), total annual cost (*TAC*), and heat exchanger surface area (*A*), were minimized separately by evaluating optimal values of seven unknown variables using four different PSO-based methods. By comparing the obtained best fitness values, the improved GQPSO approach could search quickly for better global optimal solutions by preventing particles from falling to the local minimum due to its modified local attractor scheme based on the Gaussian distributed random numbers. For example, the proposed GQPSO could predict further improved best fitness values of 40% for *NEGUs*, 17% for *TAC*, and 4.5% for *A*, respectively. Consequently, the present study suggests that the improved GQPSO approach with the modified local attractor scheme can be efficient in rapidly finding more suitable solutions for optimizing the thermal-economic problem of the crossflow PFHE.

**Keywords:** plate–fin heat exchanger; thermal-economic optimization; improved Gaussian quantum-behaved particle swarm optimization; modified local attractor

**MSC:** 49N30; 65K10; 80M50; 80M60; 90C31



**Citation:** Moon, J.H.; Lee, K.H.; Kim, H.; Han, D.I. Thermal-Economic Optimization of Plate–Fin Heat Exchanger Using Improved Gaussian Quantum-Behaved Particle Swarm Algorithm. *Mathematics* **2022**, *10*, 2527. <https://doi.org/10.3390/math10142527>

Academic Editors: Yuan Dong and Yuchao Hua

Received: 11 June 2022

Accepted: 18 July 2022

Published: 20 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A crossflow plate–fin heat exchanger (PFHE) is a compact heat exchanger composed of flat plates and fins that increase the heat transfer area. Hot and cold fluid streams in passages between the plates generate high heat transfer rates. On the one hand, since a crossflow PFHE can operate with any combination of liquids or two-phase fluids, its design offers a high degree of flexibility. Thus, this type of heat exchanger is widely used in many industries, such as in aerospace, energy conversion, and utilization because of its compact size, light weight, and effectiveness [1]. On the other hand, a disadvantage of a crossflow PFHE is higher operational and installation costs than conventional heat exchangers, due to a higher level of detail during manufacture [1]. As the demand for heat exchangers becomes more complex and diversified, it is becoming more challenging to simultaneously derive optimum values of several design variables that satisfy given operational requirements by using the conventional design approach of trial-and-error. Hence, recently, meta-heuristic (or evolutionary) optimization techniques have been introduced by

many researchers to improve the efficiency and productivity of the heat exchanger design process. For example, Yousefi et al. [2] proposed an imperialist competitive algorithm (ICA) to optimize a heat exchanger. In their study, the ICA was used, in which empires compete to take over each other's colonies. Objective functions, such as minimization of the total weight and total annual cost for the heat exchanger, were optimized with the proposed algorithm. Wang and Li [3] used the cuckoo algorithm for an irreversibility analysis of PFHEs. To achieve a better optimal solution, a non-uniform mutation operator was applied to the algorithm [3]. Khosravi et al. [4] presented an adaptive multitracker optimization algorithm (AMTOA) for global optimization problems and also applied it to solve challenging chemical engineering optimization problems including designing a heat exchanger network. Naruei and Keynia [5] proposed the wild horse optimizer (WHO) algorithm and evaluated the efficiency of the proposed algorithm in solving a heat exchange network design problem. In addition, many different heuristic optimization algorithms have been successfully proposed and applied to various practical problems [6–10].

Among the various meta-heuristic optimization techniques, particle swarm optimization (PSO) is a recently preferred population-driven optimization algorithm that models social-psychological behaviors, such as those of bird flocks or individuals, and is known as a simpler and more intuitive computational technique as compared with other heuristic optimization methods [11–13]. However, a major disadvantage of the PSO algorithm is the high possibility of premature convergence to a local minimum, which leads to a failure of searching global optimal solutions. To improve this performance degradation, several variants of the PSO methods have been developed and introduced by many researchers. For example, the quantum-behaved PSO (QPSO) algorithm was proposed based on the Schrödinger equation for the state of a particle [13–16]. Cai et al. [17] proved that the QPSO algorithm could avoid the local minimum during the optimization process and could produce an outstanding performance as compared with that of the basic PSO algorithm. Since most thermal systems are multivariable or multidimensional problems, more in-depth studies on heat exchangers that utilize the QPSO scheme are still needed. There has also been discussion on convergence and accuracy issues using a random function in the QPSO scheme. Gaussian distributed random numbers were introduced by Coelho [18] and it was found that the effectiveness of the QPSO was significantly improved [18]. This Gaussian QPSO (GQPSO) algorithm utilizes the Gaussian probability distribution to allow a continuous searching process with a series of mutation operators. However, a literature review revealed that the GQPSO algorithm has not yet been applied to the optimization problems of thermal system design.

Thus, in the present study, the first aim is to propose an improved approach based on the GQPSO algorithm for optimizing thermal system design and its operational performance. To verify the improved searching capability of the proposed improved GQPSO algorithm, its optimization results are compared with those of the basic PSO and original GQPSO methods for the design optimization problem of a PFHE with the same constrained search space suggested by Zarea et al. [19]. Regarding the second aim, the proposed GQPSO algorithm is applied to the thermal-economic optimization problem of a heat exchanger for validation of its usefulness. For this purpose, a crossflow PFHE design was chosen for the multivariable thermal-economic optimization problem. Three single objective functions: the number of entropy generation units (*NEGUs*), total annual cost (*TAC*), and heat exchanger surface area (*A*), were minimized separately by evaluating optimal values of seven unknown design variables under different calculating complexities with a specified heat power range, using the basic PSO, the original GQPSO, and the improved GQPSO methods. By comparing the calculation results of each method, the better efficiency and outperformance of the improved GQPSO algorithm were verified for the present thermal-economic design problem of a crossflow PFHE. Through the above investigations, the results of the present study suggest that the improved GQPSO algorithm can be efficient for the thermal-economic design optimization problem of a crossflow PFHE.

## 2. Materials and Methods

### 2.1. Particle Swarm Optimization

The basic PSO algorithm was suggested by Kennedy and Eberhart [20], as shown in Figure 1. As seen in a schematic of the basic PSO algorithm, an unplanned distributed population of potential solutions is initially generated in the algorithm, named particle  $i$ , where the subscript  $i = 1, 2, \dots, N_p$  indicates the number of particles of a swarm or population, and each particle has multivariables  $j = 1, 2, \dots, N_v$ . The present position in the search space, i.e.,  $x_{ij}$ , can move until the calculation stops at a predefined maximum iteration  $t_{max}$ . When a particle seeks the global best position within the swarm, the remaining particles will follow it. During iterations, every particle of the swarm searches for its own best experience and updates to particle best ( $Pbest$ ). Then, the particle communicates the latest global best ( $Gbest$ ) information for all particles and moves to a new position. The velocity vector of individual particles  $v_{ij}(t + 1)$  is generated, and then,  $x_{ij}(t + 1)$  is updated as below [15,20,21]:

$$v_{ij}(t + 1) = w \times v_{ij}(t) + c_1 r_1 [p_i - x_{ij}(t)] + c_2 r_2 [p_g - x_{ij}(t)], \tag{1}$$

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1). \tag{2}$$

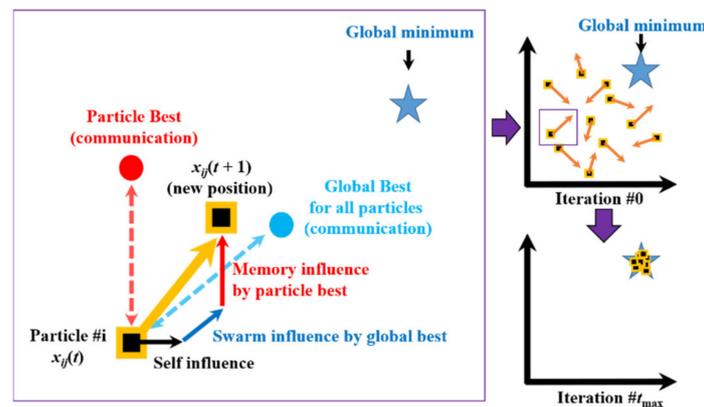


Figure 1. Concept of particle swarm optimization (PSO) algorithm.

Here,  $p_i$  is the best position for each particle ( $Pbest$ ), and  $p_g$  is the best suitable position for all particles ( $Gbest$ ), where  $p_i = [p_1, p_2, \dots, p_{N_p}]$  represents the  $Pbest$  of the  $i$ th particle,  $r_1$  and  $r_2$  are the uniformly distributed random numbers between 0 and 1,  $c_1$  is the particle cognition coefficient,  $c_2$  is the social collaboration coefficient;  $w$  is the inertia weight for the convergence and is an essential parameter to balance the exploration and exploitation behaviors between  $Pbest$  and  $Gbest$  [20], which determines the contribution rate from particle velocity to the new velocity at each iteration. Usually,  $w$  is chosen as a constant value such as 1, but the inertia weight is defined as a linearly decreasing function with time for better convergence, as below:

$$w = w_{max} - (w_{max} - w_{min}) \times \frac{t}{t_{max}}, \tag{3}$$

where  $w_{min}$  and  $w_{max}$  are the initial and final values of inertia weight, respectively, which are set as 0 and 1 in this study. The dynamic inertia weight has been proven to balance the exploration and exploitation features of the PSO methods for better performance [21,22].

### 2.2. Gaussian Quantum-Behaved Particle Swarm Optimization

A particle can exist along an absolute trajectory in conventional Newtonian mechanics, but the particle moves in a probability haze in quantum mechanics [18,23]. In the quantum world,  $x_{ij}$  and  $v_{ij}$  of a particle cannot be determined directly, but the particle has a possibility of being at the position. In the QPSO method, the state of a particle is positioned by the

wave function  $\psi[x_{ij}(t)]$  [18], instead of directly calculating the position and velocity vectors. Therefore, the behavior of the particles is dynamic in the QPSO, which is different from the basic PSO algorithm. In the QPSO algorithm, the Monte Carlo method is embedded, and thus, the particles can move according to the following iterative equation [14]:

$$\begin{cases} x_{ij}(t+1) = P_i + \beta \cdot |Mbest - x_{ij}(t)| \cdot \ln(1/u_i) & \text{if } k \geq 0.5 \\ x_{ij}(t+1) = P_i - \beta \cdot |Mbest - x_{ij}(t)| \cdot \ln(1/u_i) & \text{if } k < 0.5 \end{cases} \quad (4)$$

where  $\beta$  is the contraction–expansion coefficient, which plays a similar role as inertia weight in the basic PSO and  $u_i$  or  $k$  is a uniform probability function in the range [18]. The mean best  $Mbest$  should be defined as the mean of the  $Pbest$  for all particles and iterations as follows:

$$Mbest = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i(t), \quad (5)$$

$P_i$  is the local attractor between the global and particle bests, considering the weighted constants  $c_1$  and  $c_2$ .

$$P_i = \frac{c_1 p_i + c_2 p_g}{c_1 + c_2}. \quad (6)$$

Figure 2 shows a flowchart that compares the basic PSO algorithm with the QPSO algorithm. The initialization process is similar, but the intermediate step of the QPSO algorithm is different in that it calculates  $Mbest$  and  $P_i$  to reach an optimum solution, which are parameters that are controlled by a series of random numbers.

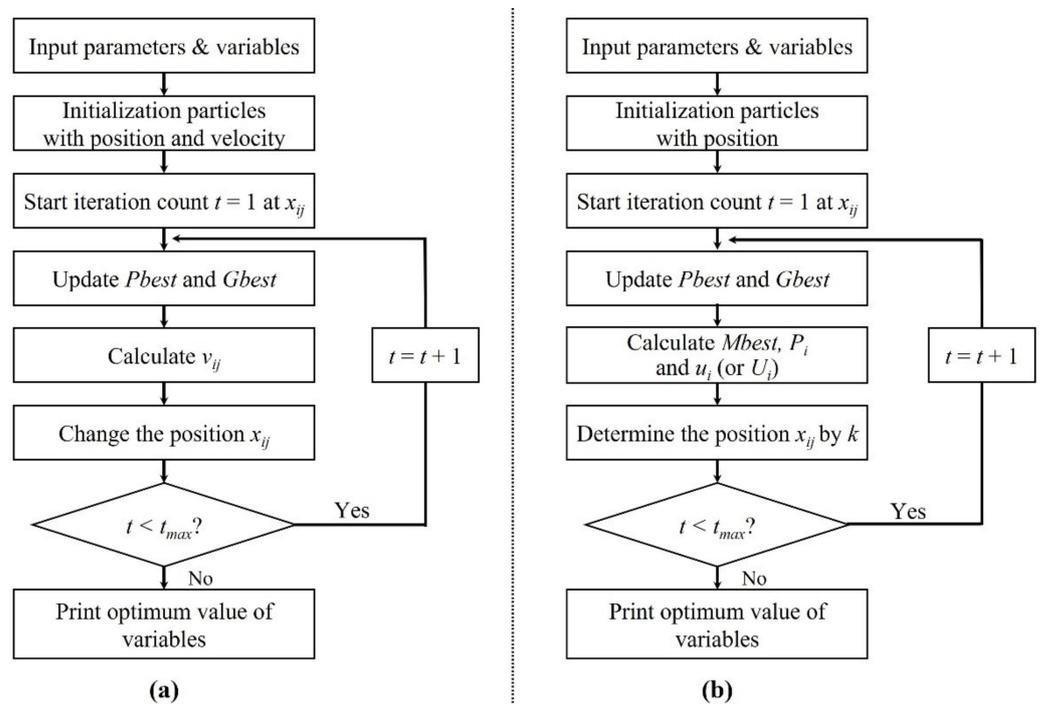


Figure 2. Flowcharts of PSO-based algorithms. (a) Basic PSO; (b) QPSO or GQPSO.

Although the QPSO algorithm can escape local minima using many random numbers, the particles may change drastically from the current position, which may lead to an unexpected divergence [18]. To reduce this unwanted divergence, the GQPSO algorithm employs the Gaussian distributed random number selection process by Coelho [18], which allows for fine-tuning due to the probability of having a large number around the current point. The GQPSO algorithm uses the absolute value of the Gaussian prob-

ability distribution with unit variance and zero mean as  $G = \text{abs}[N(0,1)]$ , where  $N$  is the unit normal distribution function. In particular, Coelho [18] introduced the following three GQPSO schemes with different mutation operators to achieve better convergence as described below:

1. Approach 1: use  $U_i = \text{abs}[N(0,1)]$  instead of  $u_i$

$$\begin{cases} x_{ij}(t+1) = P_i + \beta \cdot |Mbest - x_{ij}(t)| \cdot \ln(1/U_i) & \text{if } k \geq 0.5 \\ x_{ij}(t+1) = P_i - \beta \cdot |Mbest - x_{ij}(t)| \cdot \ln(1/U_i) & \text{if } k < 0.5 \end{cases} \quad (7)$$

2. Approach 2: use  $G = \text{abs}[N(0,1)]$  and  $g = \text{abs}[N(0,1)]$  instead of  $c_1$  and  $c_2$  at  $P_i$ ;

$$P_i = \frac{Gp_i + gp_g}{G + g} \quad (8)$$

3. Approach 3: use both Approach 1 and Approach 2.

Here,  $G$ ,  $g$ , and  $U_i$  are different Gaussian-distributed random numbers. Although the flowchart of the GQPSO algorithm illustrated in Figure 2 looks similar to the QPSO algorithm, Coelho’s work [18] revealed that the GQPSO could be a powerful strategy algorithm by employing a Gaussian mutation operator instead of the random sequences in the QPSO algorithm, and therefore, improve the search performance of optimum solutions of unknown variables by preventing premature convergence to local optima. The QPSO and GQPSO algorithms also have some disadvantages. For example, they usually require longer calculation times as compared with the basic PSO algorithm because many random numbers need to be generated during the search process for the optimum solutions. In addition, when the same constants are used, the GQPSO and QPSO algorithms tend to become more biased weighted to  $Pbest$  than to  $Gbest$ , because  $Pbest$  is used twice to calculate  $P_i$  and  $Mbest$ , as seen in Equation (7), while  $Pbest$  and  $Gbest$  of the basic PSO method balance each other by setting the same value for  $c_1$  and  $c_2$ . Thus, the GQPSO method without consideration of a proper balance of  $Pbest$  and  $Gbest$  may predict a false or local minimum. To prevent this, an improved approach based on the GQPSO algorithm is proposed to balance between  $Pbest$  and  $Gbest$  by introducing a modified local attractor  $P_i$ , as describe below:

4. Newly improved approach: use Equation (7) with  $U_i = \text{abs}[N(0,1)]$  and modified local attractor  $P_i$

$$P_i = \frac{c_1 G p_i + c_2 g p_g}{c_1 G + c_2 g} \quad (9)$$

as defined in Equation (9), the improved GQPSO approach employs random number generation based on Gaussian distribution when calculating its local attractor  $P_i$ ;  $c_1$  and  $c_2$  are the weighting constants, similar to those of the basic PSO. In addition,  $G$ ,  $g$ , and  $U_i$  are different Gaussian distributed random numbers used for the original GQPSO algorithm (Approaches 1, 2 and 3). Moreover,  $c_2$  is weighted to  $Gbest$  to balance with  $Pbest$  in the improved GQPSO, and  $c_2$  is assumed to be increased by a multiple of  $c_1$  (for example,  $c_1 = 1$  and  $c_2 = 3$ ).

### 2.3. Constraints Handling

Constraint handling techniques are applied in the proposed optimization algorithm, which involve numerical attempts to solve only the feasible space using a modified fitness function [18]:

$$\min f(x_{ij}) = \begin{cases} f(x_{ij}) & \text{if } x_{ij} \in \Omega \\ f(x_{ij}) + \text{penalty}(x_{ij}) & \text{otherwise} \end{cases} \quad (10)$$

A penalty function can be a check tool for how many feasible particles are in  $\Omega$ . In thermal system engineering problems such as the design of a crossflow PFHE,  $x_{ij}$ ,  $v_{ij}$ , and  $p_{ij}$  contain multivariable or multidimensional vectors, and, therefore, many particles can be filtered. In this study, two repair rules are considered. The first step operates to find solutions for each variable within the upper bound (UB) and lower bound (LB), that is,  $x_{ij} \in [LB_j, UB_j]$ . When  $x_{ij}$  is located beyond the two bounds, the following repair rule is applied.

$$V_j = \begin{cases} V_j = LB_j & \text{if } V_j < LB_j \\ V_j = UB_j & \text{if } V_j > UB_j \end{cases} \quad (11)$$

The second step is to check whether or not the decision variable  $g(x_{ij})$  is of bounded inequalities [18]:

$$\min f(x_{ij}) = \begin{cases} f(x_{ij}) & \text{if } g(x_{ij}) \in \Omega \\ f(x_{ij}) + q \cdot g(x_{ij}) & \text{otherwise} \end{cases} \quad (12)$$

where  $q$  is a positive constant, such as 100,000 in this study. As defined in Equation (12), the minimum objective function values of the particles existing within the feasible calculation region ( $\Omega$ ) are considered to be possible candidates for global optimal solutions ( $G_{best}$ ), while the minimum objective function values of other particles that violate the given constraints are multiplied by an arbitrary large constant ( $q$ ) to be excluded as possible candidates for  $G_{best}$ . As a result, proper values of  $G_{best}$  can be attained only among the feasible particles. Although a value of 5000 for  $q$  has been suggested in a previous study [18], a much higher number (100,000) was chosen to definitely exclude the particles beyond the given constraint region.

### 3. Crossflow Plate–Fin Heat Exchanger

In the present study, thermal design optimization is investigated with the crossflow PFHE configuration in Figure 3, which includes many alternative layers of corrugated metal fins and plates. The following assumptions are defined for comparison with other literature before the calculation of the heat exchanger:

- $N_c = N_h + 1$ , where  $N_c$  and  $N_h$  are the number of fin layers for cold and hot fluids, respectively.
- Heat exchange and heat distribution are considered uniform.
- The heat exchanger works under a steady-state.
- Longitudinal thermal resistance or heat transfer of the walls is negligible.
- The fouling or aging effect is neglected for the heat exchanger.
- The fluid physical property does not change with temperature.
- The geometry of offset-strip-fins is identical for both gases.
- Hot and cold gases are considered the ideal gases.

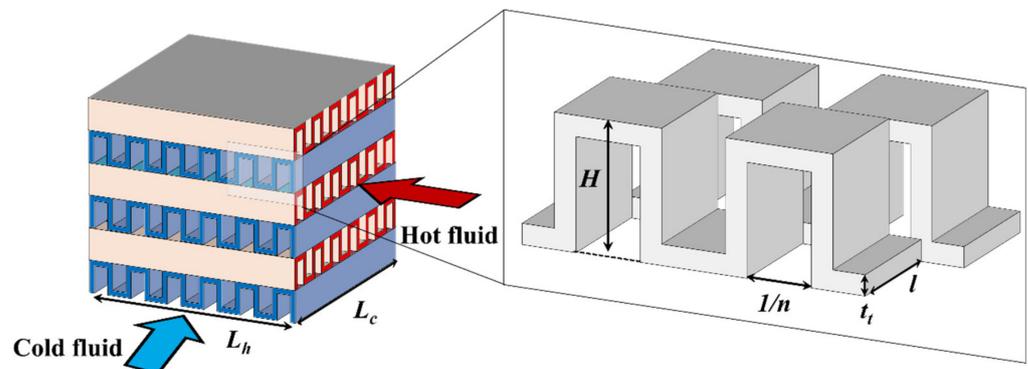


Figure 3. Schematic of corrugated crossflow PFHE.

The present optimization study employs the  $\epsilon$ - $NTU$  method to investigate the heat exchanger's efficiency in the modeling. The well-known effectiveness of the crossflow PFHE is used for two unmixed fluids [24]:

$$\epsilon = 1 - \exp \left\{ \left( \frac{1}{C_{p,r}} \right) NTU^{0.22} \left[ \exp \left( -C_{p,r} NTU^{0.78} \right) - 1 \right] \right\}, \tag{13}$$

where  $C_{p,r} = C_{p,min} / C_{p,max}$ , and the number of transfer units ( $NTU$ ) is defined as follows:

$$\frac{1}{NTU} = C_{p,min} \left[ \frac{1}{(hA)_h} + \frac{1}{(hA)_c} \right], \tag{14}$$

The heat transfer coefficient can be described in terms of the Colburn factor  $j$ :

$$h = jC_p J Pr^{-(2/3)}, \tag{15}$$

where  $Pr$  is the Prandtl number. The mass flux velocity  $J$  can be expressed as  $\dot{m} / A_{ff}$  where  $A_{ff}$  is the free flow area for hot and cold sides as below:

$$A_{ff,h} = (H - t_t)(1 - nt_t)L_c N_h, \tag{16}$$

$$A_{ff,c} = (H - t_t)(1 - nt_t)L_h N_c, \tag{17}$$

Therefore, the heat exchanger surface area for both hot and cold sides,  $A (= A_h + A_c)$ , can be calculated in Equations (18) and (19)

$$A_h = L_h L_c N_h [1 + 2n(H - t_t)], \tag{18}$$

$$A_c = L_h L_c N_c [1 + 2n(H - t_t)], \tag{19}$$

Bejan [25] suggested the entropy generation rate for the crossflow PFHE based on the gas pressure and temperature:

$$\dot{S} = \dot{m}_h \left[ C_{p,h} \ln \frac{T_{h,out}}{T_{h,in}} - R_h \ln \frac{P_{h,out}}{P_{h,in}} \right] + \dot{m}_c \left[ C_{p,c} \ln \frac{T_{c,out}}{T_{c,in}} - R_c \ln \frac{P_{c,out}}{P_{c,in}} \right], \tag{20}$$

where subscripts *in* and *out* stand for the inlet and outlet, respectively. The effectiveness of the crossflow PFHE can be arranged as:

$$\epsilon = \frac{C_{p,c}(T_{c,out} - T_{c,in})}{C_{p,min}(T_{h,in} - T_{c,in})}, \tag{21}$$

As the effectiveness of the crossflow PFHE is determined, the outlet temperatures can be obtained as well as the total heat transfer rate in Equations (22) and (23):

$$T_{h,out} = T_{h,in} - \epsilon C_{p,min} / C_{p,h} (T_{h,in} - T_{c,in}), \tag{22}$$

$$Q = \epsilon C_{p,min} (T_{h,in} - T_{c,in}), \tag{23}$$

In addition, the exit pressure for hot and cold sides with frictional pressure drops can be estimated:

$$P_{h,out} = P_{h,in} - \Delta P_h, \tag{24}$$

$$P_{c,out} = P_{c,in} - \Delta P_c, \tag{25}$$

$$\Delta P_h = \frac{4f_h L_h J_h^2}{2\rho_h d}, \tag{26}$$

$$\Delta P_c = \frac{4f_c L_c J_c^2}{2\rho_c d}. \tag{27}$$

In the offset-strip-fin configurations, the same  $f$  and  $j$  factors in Equations (28) and (29) are utilized as Zarea et al. [19] for the proper validation and optimization process:

$$f = 9.624 \cdot Re^{0.7422} \alpha^{-0.1856} \delta^{0.3053} \gamma^{-0.2659} \cdot (1 + 7.7 \cdot 10^{-8} Re^{4.429} \alpha^{0.920} \delta^{3.767} \gamma^{0.236})^{0.1}, \quad (28)$$

$$j = 0.652 \cdot Re^{-0.5403} \alpha^{-0.1541} \delta^{0.1499} \gamma^{-0.0677} \cdot (1 + 5.3 \cdot 10^{-5} Re^{1.34} \alpha^{0.504} \delta^{0.456} \gamma^{-1.055})^{0.1}, \quad (29)$$

where the following set of ratios describing the offset-strip-fin geometry should be used:

$$\alpha = s / (H - t_t), \delta = t_t / l, \gamma = t_t / s, \text{ and } s = (1/n) - t_t, \quad (30)$$

The Reynolds number can be defined using Equation (31) with a hydraulic diameter in Equation (32):

$$Re = \frac{Jd}{\mu} = \frac{\dot{m}d}{A_{ff}\mu'}, \quad (31)$$

$$d = \frac{4sl(H - t_t)}{2[sl + (H - t_t)l + (H - t_t)t_t] + t_t s}. \quad (32)$$

The number of entropy generation units, abbreviated as the *NEGUs*, represents how the heat transfer increases with decreased frictional pressure drop and irreversibility [25]. Therefore, the number of entropy generation units gives a direct method to understand the thermodynamic efficiency of a heat exchanger as follows:

$$NEGUs = \frac{\dot{S}}{(\dot{m}C_p)_{max}}. \quad (33)$$

In addition, the heat exchanger’s total annual cost (*TAC*) is considered for thermal-economic optimization [19]. The *TAC* is the summation of investment cost ( $C_{cp}$ ) and operating cost ( $C_{op}$ ).  $C_{cp}$  includes the unit area’s construction cost, heat exchanger surface area, and annual coefficient factor, where  $C_{op}$  contains the compressor electricity price as defined below [19]:

$$TAC = C_{cp} + C_{op}, \quad (34)$$

$$C_{cp} = A_{cf} C_A A^e, \quad (35)$$

$$C_{op} = \left[ \zeta \tau \frac{\Delta P \dot{m}}{\eta \rho} \right]_h + \left[ \zeta \tau \frac{\Delta P \dot{m}}{\eta \rho} \right]_c, \quad (36)$$

where  $C_A$ ,  $e$ ,  $\zeta$ ,  $\tau$ , and  $\eta$  presents the cost per unit surface area, exponent of nonlinear increase with area, electricity price of the compressor, operation time, and compressor effectivity in order.  $A_{cf}$  indicates the annual coefficient factor as below:

$$A_{cf} = \frac{r}{1 - (1 + r)^{-y}}, \quad (37)$$

where  $r$  and  $y$  are an interest rate and a depreciation time, respectively [26–28].

## 4. Results and Discussion

### 4.1. Validation of PFHE Design Model

Before thermal-economic optimization, a validation process for the crossflow PFHE design model was conducted to determine whether the complex calculation of objective function values can correctly yield the design parameters as compared with Zarea et al.’s work [19]. Here, the maximum dimensions of the present crossflow PFHE, i.e.,  $L_h \times L_c \times H$ , were assumed to be  $1 \times 1 \times 0.01 \text{ m}^3$ . The same properties for the thermal-hydrodynamic and thermal-economic parameters were also used, as summarized in Tables 1 and 2. Table 3 shows the lower and upper bounds of the design variables to be estimated, and the validation results of the given parameters are summarized in Table 4. The objective function

values of the *NEGUs* were back-calculated and validated by substituting the optimum value in Zarea et al.'s work [19], which yielded slight relative differences within an error of 1.82%, as shown in Table 4. The main reason is a digit or precision error typically observed during numerical calculations. As these errors tend to accumulate during the numerical calculation, the relative differences between this study and that of Zarea et al. [19] can be maximized with the calculation of the *NEGUs* value. As a result, we can confirm from the present validation results that the design model of the crossflow PFHE used for the present study is correct.

**Table 1.** Properties of thermal-hydrodynamic parameters.

Parameters	Hot Side	Cold Side
Mass flux, $\dot{m}$ (kg/s)	1.66	2
Inlet Temperature, $T$ (K)	1173	473
Density, $\rho$ (kg/m <sup>3</sup> )	0.6296	0.9638
Specific Heat, $C_p$ (J/kgK)	1122	1073
Viscosity, $\mu$ (Ns/m <sup>2</sup> )	$4.01 \times 10^{-5}$	$3.36 \times 10^{-5}$
Prandtl Number, $Pr$	0.731	0.694

**Table 2.** Properties of thermal-economic parameters.

Parameters	Value
Cost per Unit Area, $C_A$ (\$/m <sup>2</sup> )	90
Electricity Price, $\zeta$ (\$/MWh)	20
Operation Hours, $\tau$ (hr)	5000
Exponent of Nonlinear Increase with Area, $e$	0.6
Depreciation Time, $y$ (yr)	10
Compressor Efficiency, $\eta$ (%)	60
Interest Rate, $r$	0.1

**Table 3.** Lower and upper bounds of the design variables.

Parameters	Lower Bound (LB)	Upper Bound (UB)
Hot Flow Length, $L_h$ (m)	0.1	1
Cold Flow Length, $L_c$ (m)	0.1	1
Fin Height, $H$ (mm)	2	10
Fin Thickness, $t_f$ (mm)	0.1	0.2
Fin Frequency, $n$ (fin/m)	100	1000
Lance Length, $l$ (mm)	1	10
Number of Fin Layers at the Hot Fluid, $N_h$	1	200

**Table 4.** Validation results of crossflow PFHE design model.

Parameters	Zarea et al. [19]	Present Study	Relative Difference (%)
$\gamma$	0.346	0.346	-
$\alpha$	0.016	0.016	-
$\delta$	0.052	0.052	-
$\Delta P_h$ (Pa)	920	918	0.22
$\varepsilon$ (%)	87.0	86.8	0.23
<i>NEGUs</i>	0.1176	0.1155	1.82

#### 4.2. Thermal-Economic Optimization of PFHE Using Improved GQPSO

In this study, the same design of the crossflow PFHE was chosen as considered in Zarea et al.'s work [19] to verify the optimization performance of the proposed improved GQPSO (see Figure 3). For this, all the PSO-based algorithms, including the original and improved GQPSO approaches, were developed using the MATLAB compiler, and the

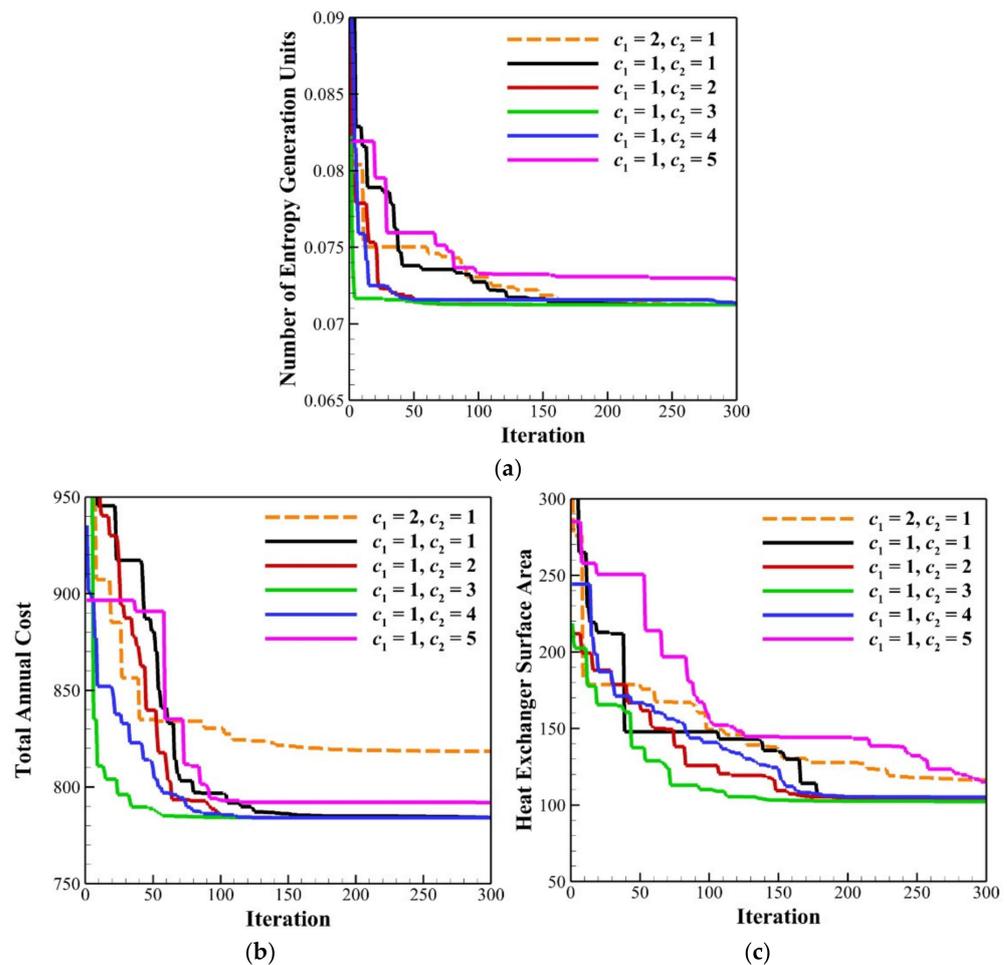
numerical calculations were performed using Intel® Core™ i7 CPU. Since all the PSO-based algorithms use random numbers and their fitness values of the objective functions vary for each calculation, the optimization was performed for at least 30 runs for each calculation.

For the first result, the influences of the particle and iteration numbers were investigated on the improved GQPSO approach with  $c_1 = 1$  and  $c_2 = 1$  to set up the optimal computing conditions. The calculation was set to be terminated when the iteration reached the predefined maximum number or when the relative difference in  $G_{best}$  values between two recent iterations became less than  $1.0 \times 10^{-8}\%$  during at least 30 consecutive iterations. The comparison results of the best and average fitness values of a single objective function, the  $NEGUs$ , with different particles and iteration numbers are summarized in Table 5. On the one hand, it is clear that smaller fitness values of the  $NEGUs$  were obtained as the number of particles/iterations increased to 100/300 and 300/300 cases when the improved GQPSO approach was used. On the other hand, relatively larger fitness values of the  $NEGUs$  were predicted with the 100/100 combination of particle size/iteration numbers. The results also show that the standard deviation of the fitness values for 30 different runs tends to decrease inversely proportional to the particle size/iteration number. For example, for the 100/300 combination, only one run fell into a local minimum among 30 runs, while the remaining 29 runs converged successfully to a global solution. In particular, the converging consistency was significantly improved depending on the iteration number rather than the particle size. The best, average, and total calculation times are summarized in Table 5 with different particle sizes/iteration numbers. Here, the best calculation time was observed for the best fitness value. The results in Table 5 reveal that a much longer calculation time was consumed, especially as the particle size increased. Based on these results, the 100/300 combination was chosen for the particle size/iteration numbers for the proposed GQPSO algorithm, considering an overall balance between the calculation time and the convergence consistency.

**Table 5.** Influences of the particle/iteration numbers on improved GQPSO approach.

Parameters	100/100	300/100	100/300	300/300
Best Fitness Value ( $NEGUs_{best}$ )	0.0717	0.0712	0.0712	0.0712
Average Fitness Value ( $NEGUs_{avg}$ )	0.0719	0.0713	0.0712	0.0712
Standard Deviation of Fitness Values ( $NEGUs_{stddev, \%}$ )	1.069	0.0112	0.01	$4.3 \times 10^{-4}$
Best Calculation Time (s)	0.77	2.37	1.84	7.40
Average Calculation Time (s)	0.85	2.66	1.97	8.03
Total Calculation Time (s)	24.7	78.0	58.4	241.5

For the second result, to find an optimal value of the social collaboration coefficient and investigate its influence on global best in the GQPSO algorithm,  $c_2$  was varied from 1 to 5, whereas  $c_1$  was fixed as 1. For this, three single objective functions: the  $NEGUs$  in Equation (33),  $TAC$  in Equation (34), and  $A$  in Equations (18) and (19), were chosen to be minimized separately for thermal-economic optimization criteria of the crossflow PFHE. Ranges of their constraint values were set to  $0.134 < \alpha < 0.997$ ,  $0.012 < \delta < 0.048$ , and  $0.041 < \gamma < 0.121$ , and  $120 < Re < 10^4$  based on previous study [19]. To achieve their minimum values, the newly improved GQPSO approach was employed. The influences of  $c_2$  on the convergence processes of the different objective functions are presented in Figure 4a–c.



**Figure 4.** Converging influence of  $c_2$  on improved GQPSO for different objective functions. (a) Number of entropy generation units (NEGUs); (b) total annual cost (TAC); (c) heat exchanger surface area ( $A$ ).

Here, the best converging results among 30 runs are depicted for each case. In Figure 4a, all the NEGUs values with different  $c_2$  values reached the global minimum before 300 iterations. The improved GQPSO approach tended to converge faster during initial iterations with higher  $c_2$  values because they increased the possibility of approaching the adjacent region of the global optimum. However, the converging speed became much slower when a  $c_2$  value much higher than 4 was used, due to a decrease in the local fine-tuning capability near the global optimum region. For example, the fastest convergence occurred with  $c_1 = 1$  and  $c_2 = 3$  after about 67 iterations, while the slowest convergence occurred with  $c_1 = 1$  and  $c_2 = 5$ . The other objective function results of Figure 4b,c indicated that the best converging performance was verified when  $c_1 = 1$  and  $c_2 = 3$  were used in the improved GQPSO approach. Based on these results, it was concluded that a better balance between  $P_{best}$  and  $G_{best}$  of the improved GQPSO approach could be attained when using the  $c_1 = 1$  and  $c_2 = 3$  combination among others. Thus, this combination was chosen for the proposed GQPSO algorithm to search optimal design parameters of the given crossflow PFHE for the best convergence performance.

For the next results, to verify the improved performance of the newly proposed GQPSO algorithm, the overall optimization results of the proposed improved GQPSO algorithm were compared with those by Zarea et al.’s work [19], the basic PSO algorithm, and the original GQPSO algorithm, as summarized in Figure 5 and Table 6. For this, the NEGUs were considered to be a single objective function to be minimized for the thermal-economic optimization criterion of the crossflow PFHE, and the best converging results among 30 runs were plotted for each method. The design of the crossflow PFHE is optimized, as shown in

Figure 3, and Table 6 reveals that the BAHPSO method of Zarea et al. [19] and the basic PSO predicted a similar order of the best fitness values of the *NEGUs*, i.e., 0.1176 and 0.1194, respectively. As shown in Figure 5, the best fitness levels of the *NEGUs* with both methods resulted in premature convergence during the remaining iterations, which reveals that the BAHPSO and the basic PSO methods fell into a local minimum; thus, their particles did not continue to search for better optimum solutions. In addition, the two GQPSO approaches (original and improved) estimated better fitness values with a similar order of 0.07, as seen in Figure 5. However, the improved GQPSO approach with  $c_1 = 1$  and  $c_2 = 3$  showed much faster convergence than the other methods. It quickly decreased to the best fitness level of the *NEGUs* during less than the initial 10 iterations, and then continued to converge gradually to the global optimal solution, 0.0712, while the original GQPSO (Approach 1) showed slower convergence to 0.0730. Furthermore, the improved GQPSO estimated about a 40% lower best fitness value than that of the study by Zarea et al. [19] and the basic PSO methods. To investigate the reason, a number of penalty particles that violated the given constraints were also checked for each algorithm, as shown in Table 6.

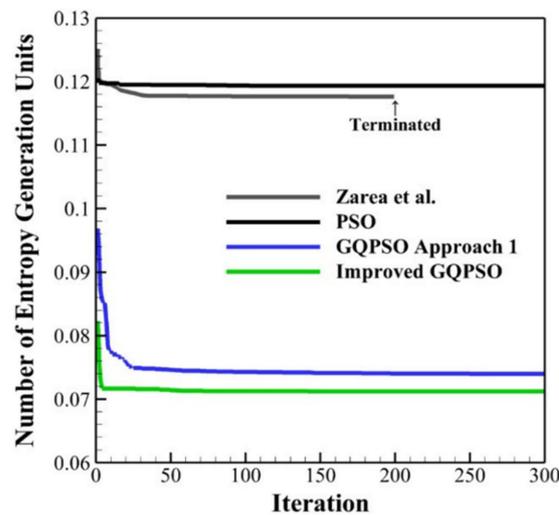


Figure 5. Comparison of converging performance between different PSO methods [19].

Table 6. Converging performance results of different PSO methods.

Parameters	Zarea et al. [19]	Basic PSO	GQPSO (Approach 1)	Improved GQPSO
Best Fitness Value ( $NEGUs_{best}$ )	0.1176	0.1194	0.0730	0.0712
Average Fitness Value ( $NEGUs_{avg}$ )	-	0.1207	0.0737	0.0712
Standard Deviation of Fitness Values ( $NEGUs_{stdev}$ , %)	-	1.95	1.52	0.0005
Best Calculation Time (s)	-	1.60	1.75	1.61
Average Calculation Time (s)	-	1.63	2.03	1.88
Total Calculation Time (s)	-	48.2	60.3	55.0
Maximum Number of Penalty Particles	-	98	94	84
Minimum Number of Penalty Particles	-	44	0	0
Average Number of Penalty Particles	-	73	37	32

A number of penalty particles that violated the given constraints were also checked for each algorithm, as shown in Table 6. In the case of the basic PSO method, the number of penalty particles varied from 44 to 98, due to the simultaneous application of several constraints. However, the improved GQPSO had the smallest average number of penalty particles, i.e., 32, during the optimization process with its superior fine-tuning capability, while 37 penalty particles were counted for the GQPSO (Approach 1). In some runs, it was all 100 particles of the improved GQPSO search for the optimum solutions within the

feasible constraints. This also resulted in the improved GQPSO providing the smallest standard deviation of the fitness values for 30 different runs. Thus, it was confirmed that the improved GQPSO provided more consistent and more reliable optimum solutions than the basic PSO and original GQPSO methods. However, Table 6 depicts that the improved GQPSO algorithm requires a slightly longer computing time than the basic PSO algorithm because more generations of random numbers are necessary for the GQPSO approach.

In particular, for the best-run case, the improved GQPSO algorithm required a short calculation time as compared to the basic PSO method with a faster convergence rate, as the particles of the improved GQPSO algorithm reached the global minimum at the early iterations. These results verify that the proposed approach using the improved GQPSO algorithm with the new local attractor scheme in Equation (9) helps prevent particles moving toward  $G_{best}$ . Since a proper balance between  $P_{best}$  and  $G_{best}$  of the improved GQPSO algorithm could minimize the number of penalty particles that lay beyond the upper and lower bounds, this led directly to rapid finding of global optimal solutions, as shown in Figure 5, within relatively short calculation time as compared with the basic PSO method.

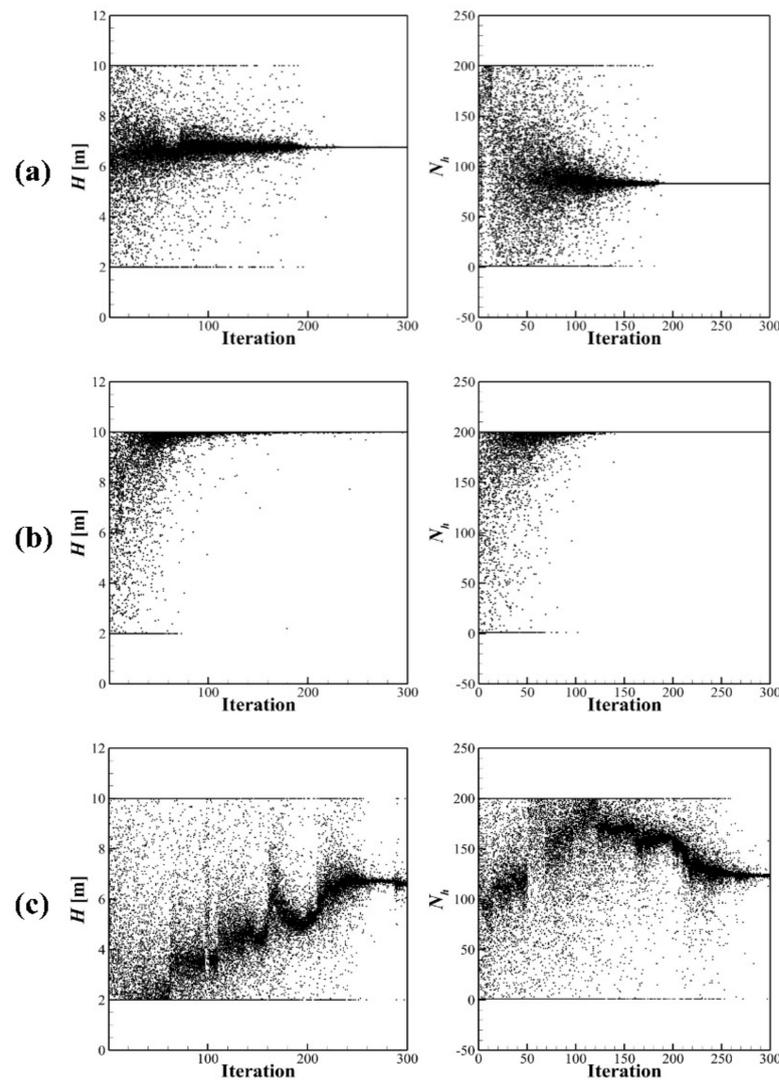
For the fourth results, the optimized design variables of the given crossflow PFHE estimated from the proposed improved GQPSO approach were compared with those by Zarea et al.'s work [19], the basic PSO algorithm, and the original GQPSO algorithm. The design variables to be optimized in this study consisted of seven variables by separately minimizing the three single objective functions: the  $NEGUs$ ,  $TAC$ , and  $A$ . First, the optimal values of design variables according to the best minimum fitness values of the  $NEGUs$  are summarized in Table 7. All the estimated data are within the given upper and lower bounds of the design variables. As compared with Zarea et al.'s work [19], the proposed improved GQPSO algorithm estimated smaller dimensions of the individual rectangular offset strip fin, leading to an increment in the number of fin layers at the hot side,  $N_h$ . Moreover, a higher effectiveness of 95.2% was predicted with the new improved GPQSO scheme when the  $NEGUs_{best} = 0.0712$ , while Zarea et al.'s work [19] achieved 87.0% as  $\varepsilon$  with  $NEGUs_{best} = 0.1176$ , as seen in Table 7.

As a result, a considerable decrement of about 40% in the values of the  $NEGUs$  was obtained with the new improved GPQSO method, which was better than those of Zarea et al.'s work [19] based on the increment in efficiency. On the one hand, Table 7 shows that the new improved GPQSO algorithm predicts further improved best fitness values of about 17% for  $TAC$  and 4.5% for  $A$ , which are lower than those of Zarea et al.'s work [19]. On the other hand, the basic PSO algorithm and the original GQPSO algorithm (Approach 1) estimate poorer best fitness values of the  $NEGUs$ ,  $TAC$ , and  $A$  objective functions than the improved GQPSO scheme. It was also observed that the basic PSO estimated a considerable number of optimal solutions at the upper and lower bound values, while the original GQPSO (Approach 1) searched optimized design variables within given bounds. However, those values of the original GQPSO did not guarantee that it could attain better fitness values or better optimization than the  $NEGUs_{best}$ ,  $TAC_{best}$ , and  $A_{best}$  of other methods. This is because the original GQPSO algorithm (Approach 1) used constant values of  $c_1$  and  $c_2$  for a local attractor in Equation (6), whereas the improved GQPSO algorithm employed random number generation using Gaussian distribution for its local attractor, as defined in Equation (9). Thus, the improved GQPSO algorithm provides improved escaping capability from the local minimum as compared with the original GQPSO algorithm (Approach 1) by balancing between  $P_{best}$  and  $G_{best}$  with the modified local attractor.

**Table 7.** Comparison of optimal design variables for different PSO algorithms.

Parameters for <i>NEGUs</i>	Zarea et al. [19]	Basic PSO	GQPSO (Approach 1)	Improved GQPSO
Hot Flow Length, $L_h$ (m)	1	1	0.9994	1
Cold Flow Length, $L_c$ (m)	0.999	1	0.7778	1
Fin Height, $H$ (mm)	7.03	10	6.80	6.76
Fin Thickness, $t_f$ (mm)	0.129	0.2	0.1004	0.1079
Fin Frequency, $n$ (fin/m)	397.3	304.5	998.8	1000
Lance Length, $l$ (mm)	7.98	10	1.19	1
Number of Fin Layers at the Hot Fluid, $N_h$	66	200	108	83
Effectiveness, $\epsilon$ (%)	87.0	86.0	94.9	95.2
Best Fitness Value, $NEGUs_{best}$	0.1176	0.1194	0.0730	0.0712
Parameters for <i>TAC</i>	Zarea et al. [19]	Basic PSO	GQPSO (Approach 1)	Improved GQPSO
Hot Flow Length, $L_h$ (m)	0.8954	0.3266	0.3108	0.3076
Cold Flow Length, $L_c$ (m)	0.9988	0.4112	0.3928	0.3884
Fin Height, $H$ (mm)	0.9977	10	10	10
Fin Thickness, $t_f$ (mm)	0.1929	0.2	0.2	0.2
Fin Frequency, $n$ (fin/m)	216	470	445	440
Lance Length, $l$ (mm)	0.9635	10	10	10
Number of Fin Layers at the Hot Fluid, $N_h$	71	200	200	200
Effectiveness, $\epsilon$ (%)	82.1	83.4	82.1	81.8
Best Fitness Value, $TAC_{best}$ (\$/yr)	939	868	798	784
Parameters for <i>A</i>	Zarea et al. [19]	Basic PSO	GQPSO (Approach 1)	Improved GQPSO
Hot Flow Length, $L_h$ (m)	0.2099	0.1851	0.3378	0.1703
Cold Flow Length, $L_c$ (m)	0.2211	0.1851	0.3641	0.1733
Fin Height, $H$ (mm)	6.7	6.8	5.3	6.6
Fin Thickness, $t_f$ (mm)	0.107	0.1	0.116	0.108
Fin Frequency, $n$ (fin/m)	1000	990.7	878.7	1000
Lance Length, $l$ (mm)	2.24	1	7.49	1
Number of Fin Layers at the Hot Fluid, $N_h$	81	110	56	123
Effectiveness, $\epsilon$ (%)	81.8	82.3	82.0	81.8
Best Fitness Value, $A_{best}$ (m <sup>2</sup> )	107.2	110.4	140.4	101.6

Lastly, the solution searching process of the improved GQPSO particles for different design variables is depicted in Figure 6 during iterations for the three single objective functions: the *NEGUs*, *TAC*, and *A*. When the calculation started, the improved GQPSO particles spread uniformly over the search domain between the upper and lower bounds. Depending on the objective function to be optimized, the improved GQPSO algorithm estimated the final optimum solutions at the given bound, as in Figure 6b for the *TAC* case, or kept finding further improved solutions and converged to them strongly, which showed significant variations during the converging process due to multiple combinations of random functions,  $G$ ,  $g$ , and  $U_i$  of Equations (7) and (9), as seen from Figure 6c for the case of *A*. While most of the basic PSO particles got closer to either the upper or lower bounds, the improved GQPSO particles tried to find and converge to a certain value between the two bounds, which indicated the enhanced search capability of the proposed approach using the improved GQPSO algorithm with local fine-tuning due to the modified local attractor scheme. The given results confirm that the proposed approach with the improved GQPSO algorithm successfully predicts better optimal design solutions that satisfy minimization of the three single objective functions: *NEGUs*, *TAC*, and *A*. Consequently, it can be concluded that the improved GQPSO algorithm with the modified local attractor outperforms other methods by quickly searching for improved global optimum solutions.



**Figure 6.** Solution searching process for improved GQPSO particles. (a)  $NEGUs_{best}$ ; (b)  $TAC_{best}$ ; (c)  $A_{best}$ .

### 5. Conclusions

In the present study, an improved GQPSO algorithm was proposed for the thermal-economic optimization problem of a PFHE. First, to verify the improved searching capability of the proposed improved GQPSO algorithm, its optimization results were compared with those of the basic PSO and original GQPSO methods for the design optimization problem of a PFHE with the same constrained search space. From the first results, it was verified that the improved GQPSO could more rapidly search global optimal solutions than other methods due to its modified local attractor scheme based on Gaussian distributed random numbers.

Next, the proposed improved GQPSO algorithm was applied to a thermal-economic optimization problem of a PFHE for validation of its usefulness. Three single objective functions: the number of entropy generation units, total annual cost, and heat exchanger surface area, were minimized separately by evaluating optimal values of seven unknown design variables using the basic PSO, original GQPSO, and improved GQPSO algorithms. By comparing the calculation results of each method, the outstanding performance of the proposed improved GQPSO algorithm could be verified for the presented thermal-economic design problem of a crossflow PFHE by preventing particles from falling to the local minimum due to a proper balance between  $P_{best}$  and  $G_{best}$  values.

Consequently, the present study suggests that the improved GQPSO algorithm with the modified local attractor scheme can efficiently and more rapidly search for more accurate global optimal solutions for optimizing the thermal-economic design problem of a crossflow PFHE. In addition, we expect that this improved GQPSO algorithm should be useful to related industries and engineers for optimizing thermal-economic designs and the operational performances of various heat exchangers.

**Author Contributions:** Conceptualization, J.H.M. and K.H.L.; methodology, J.H.M. and K.H.L.; software, J.H.M.; validation, J.H.M. and K.H.L.; formal analysis, J.H.M.; investigation, J.H.M. and K.H.L.; resources, K.H.L.; data curation, J.H.M.; writing—original draft preparation, J.H.M.; writing—review and editing, K.H.L., H.K. and D.I.H.; visualization, J.H.M.; supervision, K.H.L.; project administration, K.H.L.; funding acquisition, J.H.M., K.H.L. and D.I.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Korea Aerospace Research Institute funded by the Ministry of Science and ICT (MSIT) (NRF-2020M1A3A4A06103724), and by the National Research Foundation of Korea funded by the MSIT (NRF-2021R1F1A1049282).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

AMTOA	adaptive multitracker optimization algorithm
<i>G<sub>best</sub></i>	global best
GQPSO	Gaussian Quantum-Behaved Particle Swarm Optimization
ICA	imperialist competitive algorithm
<i>LB</i>	lower bound
<i>M<sub>best</sub></i>	mean particle best
<i>NEGUs</i>	number of entropy generation units
<i>NTU</i>	number of transfer units
<i>P<sub>best</sub></i>	particle best
PFHE	crossflow plate–fin heat exchanger
PSO	Particle Swarm Optimization
QPSO	Quantum-behaved Particle Swarm Optimization
<i>TAC</i>	total annual cost, \$/yr
<i>UB</i>	upper bound

## Nomenclature

<i>A</i>	heat exchanger surface area, m <sup>2</sup>
<i>A<sub>cf</sub></i>	annual coefficient factor
<i>A<sub>ff</sub></i>	free flow area, m <sup>2</sup>
<i>C<sub>A</sub></i>	cost per unit surface area, \$/m <sup>2</sup>
<i>C<sub>cp</sub></i>	capital cost, \$
<i>C<sub>op</sub></i>	operating cost, \$
<i>C<sub>p</sub></i>	heat capacity, J/K
<i>C<sub>p,r</sub></i>	heat capacity ratio
<i>c<sub>1</sub></i>	particle cognition coefficient
<i>c<sub>2</sub></i>	social collaboration coefficient
<i>d</i>	hydraulic diameter, m
<i>e</i>	exponent of nonlinear increase with area
<i>f</i>	fanning factor
<i>f(x)</i>	objective function
<i>G</i>	random number by Gaussian distribution
<i>g</i>	random number by Gaussian distribution
<i>g(x)</i>	constraint function

$H$	height of the fin, m
$h$	heat transfer coefficient, $W/m^2 K$
$J$	mass flux velocity, $kg/m^2 s$
$j$	Colburn factor
$k$	random number
$L$	heat exchanger length, m
$l$	fin length of the fin, m
$\dot{m}$	mass flux, $kg/s$
$N$	normal distribution function
$N_c$	number of fin layers at the cold fluid
$N_h$	number of fin layers at the hot fluid
$N_p$	number of particles
$N_v$	number of variables
$n$	fin frequency, fin/m
$P_c$	pressure at the cold side, Pa
$P_h$	pressure at the hot side, Pa
$P_i$	local attractor, m
$p_g$	global best position, m
$p_i$	particle best position, m
$Pr$	Prandtl number
$Q$	heat power, W
$q$	constant
$R$	ideal gas constant, $J/kgK$
$Re$	Reynolds number
$r$	interest rate
$rand$	random function
$r_1$	random number
$r_2$	random number
$\dot{S}$	entropy generation rate, $W/mK$
$s$	fin spacing, m
$T$	temperature, K
$t$	iteration
$t_{max}$	maximum iteration
$t_t$	thickness of the heat exchanger, m
$U_i$	random number by Gaussian distribution
$u_i$	random number
$V_j$	variable
$v_{ij}$	particle velocity, m
$w$	inertia weight
$x_{ij}$	particle position, m
$y$	depreciation time, yr

**Subscripts**

$avg$	average for different runs
$best$	best fitness function for different runs
$c$	cold side
$h$	hot side
$i$	$i^{th}$ particle
$in$	inlet
$j$	variable
$max$	maximum
$min$	minimum
$out$	outlet
$stdev$	standard deviation

**Greek Symbols**

$\alpha$	dimensionless aspect ratio for offset-strip-fin geometry
$\beta$	contraction-expansion coefficient
$\gamma$	dimensionless ratio for offset-strip-fin geometry
$\Delta P_c$	pressure drop at the cold side, Pa
$\Delta P_h$	pressure drop at the hot side, Pa

$\delta$	dimensionless ratio for offset-strip-fin geometry
$\varepsilon$	effectiveness, %
$\zeta$	electricity price, \$/Wh
$\eta$	efficiency
$\mu$	viscosity, Ns/m <sup>2</sup>
$\rho$	density, kg/m <sup>3</sup>
$\tau$	operation hours, hr
$\psi$	wave function
$\Omega$	feasible calculation region

## References

1. Taborek, J.; Hewitt, G.; Afgan, N. *Heat Exchangers—Theory and Practice*; Hemisphere Publishing: Washington, DC, USA, 1983; Volume 1.
2. Yousefi, M.; Darus, A.N.; Mohammadi, H. An imperialist competitive algorithm for optimal design of plate-fin heat exchangers. *Int. J. Heat Mass Transf.* **2012**, *55*, 3178–3185. [[CrossRef](#)]
3. Wang, Z.; Li, Y. Irreversibility analysis for optimization design of plate fin heat exchangers using a multi-objective cuckoo search algorithm. *Energy Convers. Manag.* **2015**, *101*, 126–135. [[CrossRef](#)]
4. Khosravi, R.; Khosravi, A.; Nahavandi, S.; Hajabdollahi, H. Effectiveness of evolutionary algorithms for optimization of heat exchangers. *Energy Convers. Manag.* **2015**, *89*, 281–288. [[CrossRef](#)]
5. Naruei, I.; Keynia, F. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Eng. Comput.* **2021**. [[CrossRef](#)]
6. Turgut, M.S.; Turgut, O.E. Global best-guided oppositional algorithm for solving multidimensional optimization problems. *Eng. Comput.* **2019**, *36*, 43. [[CrossRef](#)]
7. Mohanty, D.K. Application of firefly algorithm for design optimization of a shell and tube heat exchanger from economic point of view. *Int. J. Therm. Sci.* **2016**, *102*, 228–238. [[CrossRef](#)]
8. Shaukat, N.; Ahmad, A.; Mohsin, B.; Khan, R.; Khan, S.U.-D.; Khan, S.U.-D. Multiobjective core reloading pattern optimization of PARR-1 using modified genetic algorithm coupled with Monte Carlo methods. *Sci. Technol. Nucl. Install.* **2021**, *2021*, 1802492. [[CrossRef](#)]
9. Cao, Y.; Zhang, H.; Li, W.; Zhou, M.; Zhang, Y.; Chaovalitwongse, W.A. Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. *IEEE Trans. Evol. Comput.* **2019**, *23*, 718–731. [[CrossRef](#)]
10. Zhang, S.; Wu, X.; Shi, S.; Sun, X.; Christensen, R.; Yoder, G. A coupled heat transfer and tritium mass transport model for a double-wall heat exchanger design for FHRs. *Ann. Nucl. Energy* **2018**, *122*, 328–339. [[CrossRef](#)]
11. Patel, V.K.; Savsani, V.J.; Tawhid, M.A. *Thermal System Optimization*; Springer: Cham, Switzerland, 2019; Volume 1.
12. Lee, K.H. Inverse estimation of various surface emissivity distributions using repulsive particle swarm optimization. *Int. J. Heat Mass Transf.* **2019**, *134*, 1323–1332. [[CrossRef](#)]
13. Zhang, B.; Qi, H.; Sun, S.-C.; Ruan, L.-M.; Tan, H.-P. Solving inverse problems of radiative heat transfer and phase change in semitransparent medium by using Improved Quantum Particle Swarm Optimization. *Int. J. Heat Mass Transf.* **2015**, *85*, 300–310. [[CrossRef](#)]
14. Sun, J.; Feng, B.; Xu, W. Particle swarm optimization with particles having quantum behavior. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 325–331.
15. Turgut, O.E. Hybrid chaotic quantum behaved particle swarm optimization algorithm for thermal design of plate fin heat exchangers. *Appl. Math. Model.* **2016**, *40*, 50–69. [[CrossRef](#)]
16. Mariani, V.C.; Duck, A.R.K.; Guerra, F.A.; dos Santos Coelho, L.; Rao, R.V. A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. *Appl. Therm. Eng.* **2012**, *42*, 119–128. [[CrossRef](#)]
17. Cai, Y.; Sun, J.; Wang, J.; Ding, Y.; Tian, N.; Liao, X.; Xu, W. Optimizing the codon usage of synthetic gene with QPSO algorithm. *J. Theor. Biol.* **2008**, *254*, 123–127. [[CrossRef](#)]
18. Dos Santos Coelho, L. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert. Syst. Appl.* **2010**, *37*, 1676–1683. [[CrossRef](#)]
19. Zarea, H.; Kashkooli, F.M.; Soltani, M.; Rezaeian, M. A novel single and multi-objective optimization approach based on bees algorithm hybrid with particle swarm optimization (BAHPSO): Application to thermal-economic design of plate fin heat exchangers. *Int. J. Therm. Sci.* **2018**, *129*, 552–564. [[CrossRef](#)]
20. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
21. Too, J.; Abdullah, A.R.; Mohd Saad, N. A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection. *Informatics* **2019**, *6*, 21. [[CrossRef](#)]
22. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.

23. Gupta, A.K.; Kumar, P.; Sahoo, R.K.; Sahu, A.K.; Sarangi, S.K. Performance measurement of plate fin heat exchanger by exploration: ANN, ANFIS, GA, and SA. *J. Comput. Des. Eng.* **2016**, *4*, 60–68. [[CrossRef](#)]
24. Incropera, F.P.; Dewitt, D.P.; Bergman, T.L.; Lavine, A.S.; Middleman, S. *Fundamentals of Heat and Mass Transfer*, 7th ed.; John Wiley & Sons Incorporated: Hoboken, NJ, USA, 2007; Volume 1.
25. Bejan, A. The concept of irreversibility in heat exchanger design: Counterflow heat exchangers for gas-to-gas applications. *J. Heat Transf.* **1977**, *99*, 374–380. [[CrossRef](#)]
26. Patel, V.K. An efficient optimization and comparative analysis of ammonia and methanol heat pipe for satellite application. *Energy Convers. Manag.* **2018**, *165*, 382–395. [[CrossRef](#)]
27. Patel, V.K.; Rao, R.V. Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique. *Appl. Therm. Eng.* **2010**, *30*, 1417–1425. [[CrossRef](#)]
28. Rao, R.V.; Patel, V.K. Thermodynamic optimization of cross flow plate-fin heat exchanger using a particle swarm optimization algorithm. *Int. J. Therm. Sci.* **2010**, *49*, 1712–1721. [[CrossRef](#)]