

Article

Micro-Grooved Pipe Design of Parabolic Trough by Metaheuristic Optimization: An Empirical Comparison

Valentín Osuna-Enciso ^{*,†} , Marco Pérez-Cisneros [†]  and Daniel Zaldívar-Navarro [†]

Department of Electronics, Centro Universitario de Ciencias Exactas e Ingenierías—Universidad de Guadalajara, Guadalajara 44430, JAL, Mexico; marco.perez@cucei.udg.mx (M.P.-C.); daniel.zaldivar@cucei.udg.mx (D.Z.-N.)

* Correspondence: valentin.osuna@cucei.udg.mx; Tel.: +52-(33)-1378-5900

† These authors contributed equally to this work.

Received: 25 October 2019; Accepted: 18 December 2019; Published: 17 January 2020



Abstract: Pipe design is one of the most significant research lines in the area of parabolic semi-cylindrical solar collectors. The main idea behind pipe design is to increase the capillarity angle by expanding the total area being heated, therefore boosting the work capacity of the device. Such capillarity depends on several factors, whose numerical calculations are highly complex. Moreover, some of those variables are integers, whereas some others are real; hence, it is necessary to use optimization techniques that are capable of searching in those numerical spaces. There are several optimization tools that allow individual codification as binary strings, granting the coding of integer, real, or any other, as part of the same individual. Consequently, in this paper we propose the comparison of four metaheuristics when they are utilized to maximize the capillarity angle of the pipe in a parabolic trough. Experimental results show a better performance of binary particle swarm optimization when compared against the other techniques, achieving improvements in the capillarity angle of on average 11% in comparison with a similar study.

Keywords: parabolic trough; axial micro-groove; hyper-optimization; metaheuristic algorithms; optimization

1. Introduction

The use of fossil fuels for energy production has been the origin of many improvements to human life. However, it is also the cause of several problems, such as pollution and climate change. For that reason, research efforts have been focused on the utilization of alternative energy sources, like wind [1], geo-thermal [2], sun-based [3], biomass [4], hydrogen [5], bio-fuels [6], or even tidal [7]. Contrary to fossil fuels, these are considered renewable because they are sustainable, cost-effective, and environmentally friendly [8,9]. Because of its abundance and availability in many places around the world, the sun is the most critical source. Therefore, much research has focused on improvements to the associated technologies to extract its energy, such as photo-voltaic and concentrated solar power (CSP) systems [10].

In CSP systems of the parabolic trough collector (PTC) type, heat pipes (HPs) receive the concentrated solar power to transfer the temperature into a heat transfer fluid (HTF) that is later utilized to perform mechanical work. Among all the components of PTCs, HPs are considered the most crucial element to increase the efficiency in energy harvesting [11,12] by boosting the transformation of sunlight to heat. One way to improve the receiver efficiency is by modifying its parameters. Thus, the optimal design of HPs is an essential need to be covered by research endeavors [13]. This work proposes a comparison of four metaheuristic methods when they are used to increase the capillarity angle of micro-grooved pipes.

Various components of CSP systems are feasible for optimization, such as (1) the reflector, (2) the collector, (3) the heat receiver, (4) heat transport, (5) power production, and (6) thermal storage. The use of metaheuristic algorithms is a relatively recent trend in research related to CSP improvements. In that sense, several studies have been made, particularly in the last four decades. There are two optimization methods that can be used: classical and evolutionary. The second class of techniques are the main focus of this work. For example, Wang and Yang [14] proposed the optimization a prism-based solar concentrator via genetic algorithm, achieving improvements in brightness of around 400% when compared with a similar approach. In this case, the main idea is to find the optimal combination of prismatic crystals to enhance sunlight energy transmission via optic fiber. Nazemi and Boroushaki describe the optimal design of a dish/stirling system in [15]. The model considers the parabolic dish and the cavity receiver as the optimization problem, and the metaheuristic used was particle swarm optimization. The obtained design and a real system were compared, finding that the approach gave a good approximation of real devices. In the same venue, authors in [16] optimized the coefficients of a photovoltaic/thermal system through a genetic algorithm, finding improvements in the thermal addition to its exergetic efficiency.

The mentioned articles pay attention mainly to the collector; however, the heat transmission between the collector and the HTF is another vital aspect of CSP research. In that sense, Petersont began studies related to pipes with V-shaped micro-grooves [17], although other kinds of improvements to pipes have been put forth. For instance, authors in [18] propose the insertion of twisted tapes in that kind of exchanger, with the idea of increasing the heat transfer by modifying the Reynolds and Nusselt numbers. In [19], the pipe optimization considered two criteria: (1) the investment costs, and (2) the coupling between the CSP system and the steam turbine discharge. The authors found that the use of finned hexagonal tubes is the best option to reduce cost in commercial production, packing, and storage. Another proposal is the use of helically finned tubes; for example, authors in [20] made numerical simulations for the utilization of those pipes in the design of a CSP, and they found that the whole system's efficiency could be increased with their use. A recent attempt to optimize the design of an HP using evolutionary algorithms was made in [21]. Here, the use of genetic algorithms was proposed [22] to improve the design of the absorber in a parabolic trough. Several variables were examined and used to improve the capillarity. Additionally, an adiabatic model was proposed to calculate the liquid front, as well as its establishment velocity. The authors made improvements of around 10% in comparison with another study [23].

Our proposal improves upon the results shown in [21], with different metaheuristic algorithms: discrete binary differential evolution (DBDE), clonal selection algorithm (CSA), binary particle swarm optimization (BPSO), and genetic algorithm (GA). Because of their configuration, these metaheuristic algorithms are capable of searching in mixed numerical spaces. In that sense, this work contributes, on the one hand, to the search for better algorithms [24] to solve the capillarity problem in the design of micro-grooved pipes in CSP systems. On the other hand, it contributes to the use of the capillary-driven two-phase flow model proposed by [21]. The approach obtained improved upon HP designs by maximizing the capillarity angle. Even though other fluids (e.g., vegetable oils, synthetic oil) are used frequently as HTFs for CSPs [25], in the present study only four commonly utilized fluids are considered for the design optimization of CSP systems: liquid sodium, water, and two molten salts [26]. From the compared techniques, BPSO achieved improvements in the wet front, via the capillarity angle, from around 11% against the study reported in [21]. The results suggest that metaheuristic algorithms are a good option for the optimal design of micro-grooved HPs in CSP systems. The rest of the article is structured as follows: Section 2 explains the details concerning the micro-groove pipe design, particularly its governing equations. Section 3 deals with the details regarding the utilized metaheuristic algorithms, whereas Section 4 describes the experimental part. Section 5 provides some conclusions of the present work.

2. Design of Micro-Grooved Pipe

The study made in [21,27] considered the capillarity angle and the velocity of a heat transfer fluid (HTF) inside a heat pipe (HP) in the model. This work used the same design and held the same assumptions. The mathematical explanation of the model is in the next subsection, and the other paragraph describes its solution methodology.

Governing Equations

The use of micro-grooved pipes as a heat receiver is frequent in CSP systems. Authors in [27] developed a capillary-driven two-phase model with several design variables. In the first phase, it is assumed, that there is no heat flux for the calculus of the liquid front velocity. The second stage establishes two relationships: heat flux vs. liquid front velocity, and mass of liquid vs. liquid front velocity. Those relations are crucial to calculate the fluid supply to a micro-grooved pipe (Figure 1) under the CSP working conditions.

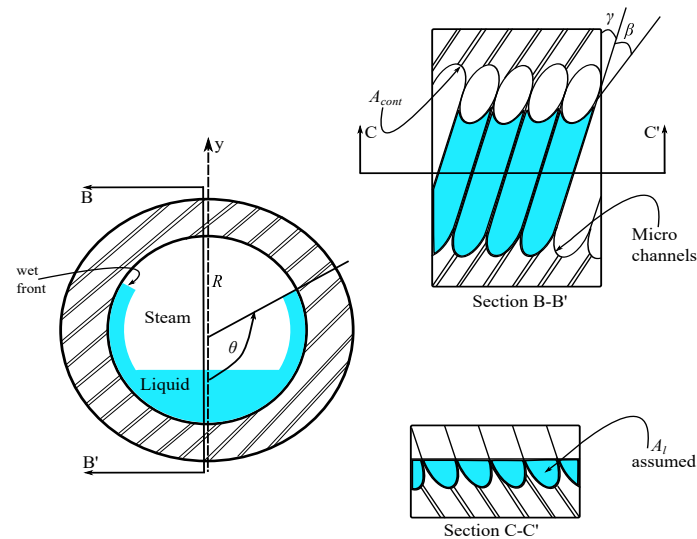


Figure 1. Pipe with micro-grooves (adapted from [21]).

It is important to remark that the model only predicts the behavior of saturated liquids with different operating temperatures. Additionally, the steam being displaced by the moving fluid in the micro-grooves is non-negligible. By considering those situations, the flow behavior in this kind of pipe is influenced only by the momentum balance of the liquid in the direction of the capillarity angle, θ , as shown in Figure 1. The momentum balance is given by:

$$F_I = F_c + F_g + F_v, \quad (1)$$

where F_I are the inertial forces over the liquid layer, F_c are the capillarity-driven forces, F_g are the gravitational forces, and F_v are the viscous forces. Other assumptions are:

- Laminar and Newtonian flux in the micro-groove;
- Isothermal layer of fluid, therefore its internal friction is negligible; and
- The thickness of the liquid layer advancing on the micro-groove insignificant.

If we consider a mass for the liquid that flows in the direction of the capillarity angle, with a liquid front velocity, then Equation (1) becomes [23]:

$$\frac{dv_l}{d\theta} = \frac{1}{\rho_l \theta v_l} \left(\frac{2\sigma_l \cos\beta}{r_c} - \rho_l g R (1 - \cos\theta) \cos\gamma - \mu_l \frac{v_l}{r_H} \frac{A_{cont}}{A_l} - v_l^2 \rho_l \right), \quad (2)$$

where the meaning of every variable and constant is in Appendix A. In this work, the objective is to maximize the capillarity angle (e.g., liquid front position). Equation (2) is prone to errors if used to calculate θ^* , the maximum capillarity angle. Therefore, authors in [23] proposed its reformulation using time as the independent variable:

$$\begin{cases} \frac{d\theta}{dt} = \frac{v_l}{R} \\ \theta \frac{dv_l}{dt} = \frac{2\sigma_l \cos\beta}{R\rho_l r_c} - g\cos\gamma(1 - \cos\theta) - \mu_l \frac{v_l}{r_H} \frac{A_{cont}}{RA_l\rho_l} - \frac{v_l^2}{R} \end{cases} \quad (3)$$

Methodology to Solve the Governing Equations

Equation (3) can be re-formulated as a system of differential algebraic equations:

$$\begin{bmatrix} 1 & 0 \\ 0 & \theta \end{bmatrix} \begin{bmatrix} \frac{d\theta}{dt} \\ \frac{dv_l}{dt} \end{bmatrix} = \begin{bmatrix} \frac{v_l}{R} \\ \frac{2\sigma_l \cos\beta}{R\rho_l r_c} - g\cos\gamma(1 - \cos\theta) - \mu_l \frac{v_l}{r_H} \frac{A_{cont}}{RA_l\rho_l} - \frac{v_l^2}{R} \end{bmatrix} \quad (4)$$

with initial conditions $\theta(t = 0)$ and $v_l(t = 0)$. In the experimental part, Equation (4) is evaluated in 10 seconds by using the Matlab solver *ode45*, to find the stable values of θ and v_l . An illustrative example of the evaluation of Equation (4) is in Figure 2, where the initial conditions are $\theta(t = 0) = 0$ and $v_l(t = 0) = 0.4$. This Figure also depicts the dynamics of the capillarity angle and the velocity of the liquid front. Concerning the angle, it surpasses the stable point around the first second, with a tendency to stability after around 1.5 s. We consider maximizing the value of the capillarity angle in the stable state of the simulation.

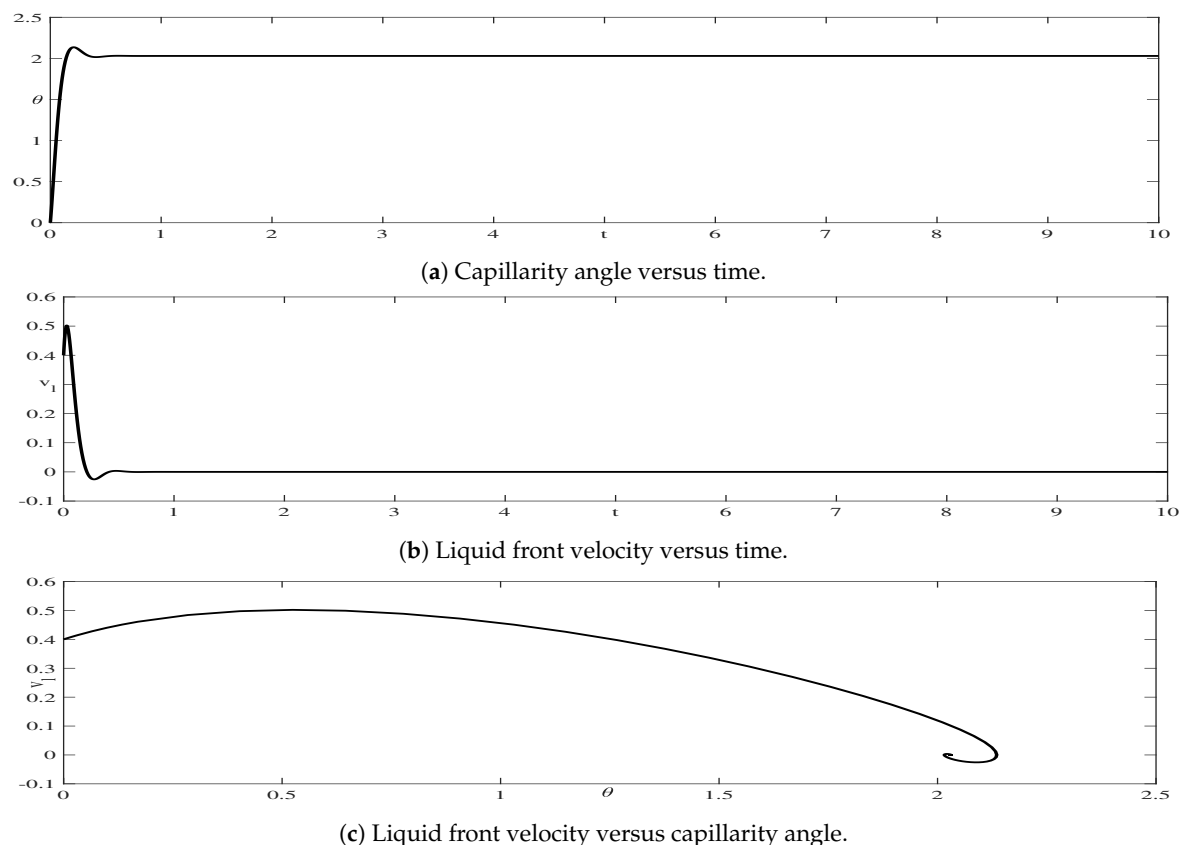


Figure 2. A simulation of Equation (4), solved with the *ode45* Matlab solver.

The optimization process produces an increase in the capillarity angle, which enhances the heat transfer to the micro-grooved pipe. Then, the objective function can be described as the maximization of the capillarity angle:

$$\max \theta = \theta(R, \beta, \gamma, st, p, \phi, r_m, kf, tg). \quad (5)$$

The value θ is function of several variables that are defined in Appendix A. In this work, four fluids (Appendix B) are used in the optimization of the capillarity angle; while some parameters depend on the kind of fluid (kf), such as liquid density, viscosity and dynamic density (Appendices C–F), other are independent of such variable, e.g. the micro-groove geometry (Appendix G).

3. Metaheuristic Algorithms

This section briefly explains the four selected metaheuristic algorithms utilized to solve the problem of pipe design in the parabolic trough. In each technique, the individual codification was as binary strings, because they are capable of encoding the two kinds of variables optimized in this article: integers and real values. Another critical remark is the fact that in every algorithm, the initialization of the population is performed by considering a uniform probability distribution [28].

3.1. Discrete Binary Differential Evolution

This motivation of this metaheuristic is the evolutionary process, similar to the genetic algorithms. Nevertheless, and even though the operators' names are the same, they function in a slightly different manner. Such operators explore the search space efficiently, allowing the algorithm to achieve the right balance between exploration and exploitation [29]. The algorithm was first proposed to minimize nonlinear and non-differentiable functions. In their seminal paper, Storn and Price considered some essential features of this optimization technique: few control variables, easily parallelizable, and excellent convergence properties, among others [28]. The basic algorithm for real candidate solutions utilizes three operators, which modify the whole population of vectors: (1) mutation, (2) crossover, and (3) re-selection of the best individual, which are mostly the same operators used in the discrete binary differential evolution (DBDE) [30] utilized in this work. A detailed description of those mathematical operators is given in the following.

Mutation

For DBDE, this operator must select three random parents from the original population with the restriction that $i \neq r_1 \neq r_2 \neq r_3$, where i is the actual individual and r_1, r_2 , and r_3 are three randomly generated integers from a uniform distribution. Once selected, the mutated vector is calculated by

$$\mathbf{V}_i = \mathbf{b}_{r_1} + F(\mathbf{b}_{r_2} - \mathbf{b}_{r_3}), \quad (6)$$

where $F \in [0, 2]$ represents the step size, and the binarization is achieved by

$$v_{i,j} = \begin{cases} 1 & \text{if } rand() \leq \frac{1}{1 + \exp(-V_{i,j})}, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

with $rand()$ being a uniform random number.

Crossover

The mutated vector, as well as the actual candidate solution, serve to construct a trial vector:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } (rand() \leq CR \parallel j == jrand), \\ b_{i,j} & \text{otherwise,} \end{cases} \quad (8)$$

where $rand()$ is a uniform random number, $CR \in [0, 1]$ is the crossover rate, $jrand \in [0, N_b]$ is an integer number generated from a uniform distribution. This value is to ensure that at least one bit is mutated.

Selection

The final step of the algorithm consists of a greedy re-selection of the objective function value of the actual candidate solution, and the fitness value of the trial vector:

$$\mathbf{b}_i = \begin{cases} \mathbf{u}_i & \text{if } (f(\mathbf{u}_i) > f(\mathbf{b}_i)), \\ \mathbf{b}_i & \text{otherwise.} \end{cases} \quad (9)$$

3.2. Clonal Selection Algorithm

The environment is a source of many health imbalances to many living beings, particularly mammals. To counteract that situation, those individuals have several internal systems, one of them being the immune system. This system has compelling attributes, such as learning capability, pattern recognition, and memory [31]. A theory formulated by immunologists is called the clonal selection theory; in this theory, the immune system's elements (antibodies) are continually monitoring for external substances which attacked the body in the past. If a component of that kind appears (e.g., an antigen), then an immune response is triggered, with the following steps: cloning, mutation, destruction of the foreign agent, re-selection, and memory. Those characteristics have served as an inspiration to computer scientists to create an area of soft computing called artificial immune systems, which has several variants. The version utilized in this paper is the clonal selection algorithm [32], whose steps are: (1) cloning, (2) hyper-mutation, (3) re-selection, and (4) diversity introduction.

Cloning and Hyper-Mutation

Even though there are other schemes, this paper uses the one related to the simple cloning of the individuals, considering a certain percentage, P_c . First, it generates a cloned population, $\mathbf{C}$, which has N_c clones

$$N_c = \sum_{i=1}^{N_p} \text{round}(P_c \cdot N_p) \quad (10)$$

and then mutation is carried out with a probability pm to every bit in the population, to get a new cloned and mutated population, $\mathbf{C}_m$

$$c_{m_{i,j}} = \begin{cases} \text{not}(c_{i,j}) & \text{if } (rand() \leq pm), \\ c_{i,j} & \text{otherwise} \end{cases} \quad (11)$$

It is essential to say that those individuals will compete against the original parents from $\mathbf{B}$. Consequently, the evaluation of clones will be in groups: the fitness value of every parent will be contrasted against the fitness of its mutated clones, repeating the procedure for each parent of the original population.

Re-Selection and Diversity Introduction

In order to evaluate each individual from $\mathbf{C}_m$, it is necessary to transform those binary strings to real and integer values, obtaining $\mathbf{x}_{mi}$:

$$\mathbf{b}_i = \begin{cases} \mathbf{c}_{mi}^{best} & \text{if } f(\mathbf{x}_{mi}) > f(\mathbf{x}_i), \\ \mathbf{b}_i & \text{otherwise,} \end{cases} \quad (12)$$

where f is the objective function, $\mathbf{x}_i$ is the real version of $\mathbf{b}_i$, $\mathbf{x}_{mi}$ are the real version of the clones corresponding to individual i , and $\mathbf{c}_{mi}^{best}$ is the best binary candidate solution extracted from the clones

corresponding to individual i . Once the re-selection between $\mathbf{C}_m$ and $\mathbf{B}$ is complete, the last phase of the CSA is a diversity introduction consisting of replacing a percentage of the worst individuals in $\mathbf{B}$ with random individuals created with Equation (22). As in other evolutionary metaheuristics, all the steps repeat until a criterion is reached, usually until a maximum iteration number.

3.3. Binary Particle Swarm Optimization

This metaheuristic takes as inspiration the behavior of flocks of bird flocks and schools of fish. It was initially proposed as an optimizer by Kennedy and Eberhart in 1995 [33]. The algorithm considers a population of individuals called particles, which represent candidate solutions for a problem (usually an optimization problem); this set also describes a swarm from the analogous point of view. In the algorithmic implementation, the particles evolve by moving around the space of solutions, considering the velocity of each particle around itself and the best global particle found so far. In that sense, the original technique has two operators: (1) the velocity calculus, which considers the global and the local social behavior of each particle, and (2) position upgrade. The next lines describe the operators mathematically for the binary particle swarm optimization (BPSO) used in this paper [34].

Velocity Calculus

In BPSO, the global velocity change utilizes two vectors which hold the probability that every bit changes either to zero or to one; the respective equations to calculate these vectors are

$$v_{i,j}^0 = w \cdot v_{i,j}^0 + d_{(i,j)_1}^0 + d_{(i,j)_2}^0, \quad (13)$$

$$v_{i,j}^1 = w \cdot v_{i,j}^1 + d_{(i,j)_1}^1 + d_{(i,j)_2}^1, \quad (14)$$

where w represents the inertia weight, and $d_{(i,j)_1}^0, d_{(i,j)_2}^0, d_{(i,j)_1}^1$, and $d_{(i,j)_2}^1$ are temporary values calculated with the following conditions:

if $b_{ibest_j} = 1$ then $d_{(i,j)_1}^1 = c_1 \cdot r_1$ and $d_{(i,j)_1}^0 = -c_1 \cdot r_1$,
 if $b_{ibest_j} = 0$ then $d_{(i,j)_1}^0 = c_1 \cdot r_1$ and $d_{(i,j)_1}^1 = -c_1 \cdot r_1$,
 if $b_{gbest_j} = 1$ then $d_{(i,j)_2}^1 = c_2 \cdot r_2$ and $d_{(i,j)_2}^0 = -c_2 \cdot r_2$,
 if $b_{gbest_j} = 0$ then $d_{(i,j)_2}^0 = c_2 \cdot r_2$ and $d_{(i,j)_2}^1 = -c_2 \cdot r_2$,

with the indexes i and j representing the actual individual and the bit, respectively; $c_1, c_2 \in [0, 2]$ are the constriction factors taken from the original PSO; r_1 and r_2 are random numbers drawn from a uniform distribution; and b_{ibest_j}, b_{gbest_j} are the bits that belong to the local and global best individuals, respectively. After calculating $v_{i,j}^0$ and $v_{i,j}^1$, the next step consists of transforming those probabilities into the velocity change of the actual particle:

$$v_{i,j}^c = \begin{cases} v_{i,j}^1 & \text{if } b_{i,j} = 0, \\ v_{i,j}^0 & \text{if } b_{i,j} = 1. \end{cases} \quad (15)$$

Position Upgrade and Selection

The position calculus of the actual particle in the next iteration is

$$b_{i,j}(k+1) = \begin{cases} b_{i,j}^-(k) & \text{if } r_1 \leq \frac{1}{1+e^{-v_{i,j}^c}}, \\ b_{i,j}(k) & \text{if } r_1 > \frac{1}{1+e^{-v_{i,j}^c}}, \end{cases} \quad (16)$$

where $b_{i,j}^-$ is the 2 complement of $b_{i,j}$, and r_1 is a uniform random number. The last stage in the BPSO is the comparison of the generated candidate solution against the local best as well as the global best:

if $f(\mathbf{x}_i) > f(\mathbf{x}_{ibest})$ then $f(\mathbf{x}_{ibest}) = f(\mathbf{x}_i)$ and $\mathbf{b}_{ibest} = \mathbf{b}_i$,

if $f(\mathbf{x}_i) > f(\mathbf{x}_{gbest})$ then $f(\mathbf{x}_{gbest}) = f(\mathbf{x}_i)$ and $\mathbf{b}_{gbest} = \mathbf{b}_i$,

where $\mathbf{x}_{ibest}$, $\mathbf{x}_{gbest}$, and $\mathbf{x}_i$ are the real versions of the binary $\mathbf{b}_{ibest}$, $\mathbf{b}_{gbest}$, and $\mathbf{b}_i$, respectively.

3.4. Genetic Algorithms

Considered as an evolutionary algorithm [22,35], this metaheuristic was proposed in the 1960s. This algorithm is based on Darwin's theory of evolution, where individuals are represented by bit strings; similar to the previous algorithms, this description is useful to codify simple data structures. As an optimizer, this technique is capable of finding competitive solutions to complex optimization problems [35]. Therefore, it has been extensively used to solve problems from different domains (machine learning, operations research, image processing, etc.), and it was even utilized in a similar study to the one presented in this article [21]. The operators used in the canonical version of the genetic algorithm (GA) are (1) interchange of genetic material, (2) a mutation scheme, and (3) re-selection of the fittest. Below is the mathematical description of these operators.

Crossover

One method to achieve crossover in GA is the roulette technique, in which it is necessary to calculate the fitness of every individual, and then the cumulative fitness as

$$E = \sum_{i=1}^{Np} f(\mathbf{x}_i), \quad (17)$$

where $\mathbf{x}_i$ is the real representation of the binary individual $\mathbf{b}_i$. Later, the selection probability as well as the cumulative selection probability of every individual are computed according to their fitness (Equations (17) and (18)):

$$p_i = f(\mathbf{x}_i) / E, \quad (18)$$

$$q_i = \sum_{j=1}^i p_j. \quad (19)$$

Two random numbers, r_1 and r_2 , from a uniform distribution are generated after the previous calculus, and the selection applies to the parents that meet $q_i < r_1$ and $q_i < r_2$. After the selection of two parents in the prior step of the GA, the next step corresponds to the crossover, which is the crossover of one point in this paper. This method works by choosing, with a given probability pc , a point of the selected parents to generate two new individuals by an interchange of genetic information. The crossover points are calculated with $r_1 \cdot Nb$ and $r_2 \cdot Nb$.

Mutation and Reselection

The mutation considers the change of every bit in the binary individual by taking the mutation probability pm :

$$bm_{1,j} = \begin{cases} b_{i,j}^- & \text{if } r_1 \leq pm, \\ b_{i,j} & \text{otherwise,} \end{cases} \quad (20)$$

where $bm_{1,j}$ is the binary and mutated child number 1; the same procedure applies for child number 2. Finally, the last step in GA is the competence between the two parents and the two children:

$$\mathbf{b}_i = \begin{cases} \mathbf{bm} & \text{if } f(\mathbf{x}_m) > f(\mathbf{x}_i), \\ \mathbf{b}_i & \text{otherwise.} \end{cases} \quad (21)$$

3.5. Population Initialization

As mentioned previously, the optimization tool utilized in the problem of pipe design in a parabolic trough must be capable of searching in two numerical spaces, namely, real, and integer. For that reason we use binary versions of the aforementioned algorithms, which allow the management of those codifications. In that sense, the conformation of each individual both as real and as integer values is

$$\mathbf{x}_i = [R_i, \beta_i, \gamma_i, st_i, p_i, \phi_i, r_{mi}, kf_i, tg_i], \quad i = 1, \dots, N_p, \quad (22)$$

whose binary representation—denoted by $\mathbf{b}_i$ —is a string of 150 bits. A more detailed explanation of each variable, their limits, as well as the number of bits used for a particular codification, is given in Table 1.

Table 1. Variables in the individual codification.

Symbol	Low Limit, l	High Limit, u	Bits	Units	Space
R	0.025	0.05	21	m	$\mathbb{R}$
β	0	1.39	21	rad	$\mathbb{R}$
γ	0	1.39	21	rad	$\mathbb{R}$
st	233	1600	21	K	$\mathbb{R}$
p	0.00025	0.0005	21	m	$\mathbb{R}$
ϕ	0.1745	1.0472	21	rad	$\mathbb{R}$
r_m	0.00025	0.0005	21	m	$\mathbb{R}$
kf	0	3	2		$\mathbb{Z}$
tg	0	1	1		$\mathbb{Z}$

In the four compared algorithms, the population is created by

$$\mathbf{b}_i = \text{hardlim}(2 \cdot \text{rand}(1, N_b) - 1), \quad i = 1, \dots, N_p, \quad (23)$$

where N_p is the population size, N_b is the total number of bits in each string, $\text{rand}(1, N_b)$ is a row vector of N_b uniformly generated random numbers, $\text{hardlim}(\cdot)$ is a positive limiting function, and l, u are respectively the lower and upper limits of the search space. Finally, Equation (5) representing the objective function can be written as:

$$f(\mathbf{x}^*) = \max_{\forall \mathbf{x} \in \mathbb{R}, \mathbb{Z}} \theta(\mathbf{x}). \quad (24)$$

4. Experimental Setup and Results

In the methodology applied in this paper, the optimization problem consists of maximizing the capillarity angle described by Equations (24) and (5). A virtual pipe was constructed to evaluate the objective function, and its behavior was simulated until 10 s (Figure 2). Once the optimization problem was defined, the next step was tuning the specific parameters that belong to every algorithm utilized in this work. To adjust the settings, we used a hyper-optimization scheme, where a statistical adjustment was employed. Finally, all the algorithms were compared under different criteria when they were utilized to solve the problem considered in this work.

4.1. Tuning

In order to make fair comparisons among the metaheuristic algorithms, we used the framework presented in [36], in which a higher-level algorithm (in this case, differential evolution) is utilized to tune every algorithm. This scheme is known as a hyper-optimization problem, where the main idea is to minimize the computational effort (represented by the number of iterations) of a given algorithm:

$$\min(t_\delta) = A(f(\mathbf{x}), \mathbf{p}), \quad (25)$$

considering that t_δ is the iteration number achieved under the restriction $|f_{\min} - f^*| \leq \delta$, f^* is the best result achieved by a similar study [21], and $f_{\min}$ is the best outcome achieved by the algorithm being tuned. Additionally, the value of the threshold δ is set to 1×10^{-5} as used in [36], $\mathbf{x}$ is the candidate solution for the problem being optimized, and $\mathbf{p}$ is the candidate parameter tuning for the algorithm A . The tuning algorithm was run 30 times for every metaheuristic reported in this work to get the statistics of the best parameters. Results are shown in Table 2 and following the format $\mu(\sigma)$. It is important to note that the vector $\mathbf{p}$ is composed of the parameters of every algorithm, and $A \in \{CSA, GA, BPSO, DBDE\}$.

Table 2. Statistical results of the parameter tuning, in the format mean(standard deviation). CSA: clonal selection algorithm; GA: genetic algorithm; BPSO: binary particle swarm optimization; DBDE: discrete binary differential evolution.

CSA		GA		BPSO		DBDE	
p_m	p_d	p_m	p_c	c_1	c_2	w	c_r
0.5583 (0.2905)	0.4383 (0.1100)	0.5816 (0.2389)	0.3517 (0.2788)	0.9020 (0.7751)	0.5425 (0.3876)	10.2175 (0.3338)	0.5025 (0.3246)

As mentioned in [36], in general terms the parameter tuning depends on the problem being solved. In that sense, after reviewing the results shown in Table 2, and based on the standard deviation, it can be considered that the most sensitive parameters—at least for the problem of the parabolic trough—are p_d , p_m , w , and c_r . A single run of every algorithm with the tuning parameters of Table 2 gave the results shown in Table 3 and Figure 3.

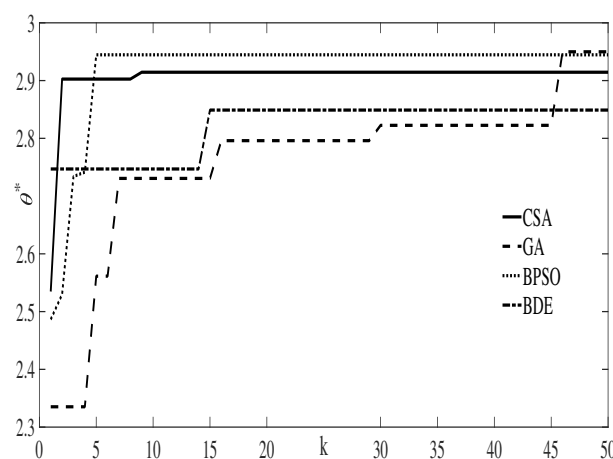


Figure 3. Convergence of capillarity angles after one run, with k being the actual iteration.

Table 3. Best individuals found after one run.

Description	Variable	CSA	GA	BPSO	BDE	Units
Pipe inner radius	R	0.0261	0.0399	0.0286	0.0316	m
Contact angle	β	0.3713	0.0501	0.7484	0.5932	rad
Micro-groove angle	γ	1.1524	1.2189	1.1008	1.1633	rad
Saturated temperature	st	426.74	425.74	463.36	454.00	K
Groove depth	p	0.0004	0.0004	0.0004	0.0004	m
Apex angle	ϕ	0.5934	0.1773	0.7957	0.9504	rad
Equivalent capillary rad	r_m	0.0003	0.0004	0.0003	0.0005	rad
Kind of fluid	kf	1	1	1	1	
Type of groove	tg	0	0	0	0	

4.2. A General Comparison

The first experimental set consisted of completing 100 tests per algorithm, for statistical purposes. The main idea was to compare their performance by considering the best design solution discovered at every run, and by acknowledging that every metaheuristic is searching in the entire design space of nine variables (Table 3). Every run had 50 iterations and the same initial population, to make a fair comparison among the techniques. The first 30 results after every trial are in Table 4, with the best result of the four metaheuristics in bold letters.

Table 4. Best capillarity angles of 30 runs. The best results of the four metaheuristics are in bold.

Run	CSA	GA	BPSO	DBDE
1	2.83750	2.89947	2.99167	2.66811
2	2.80859	2.56935	2.97142	2.65656
3	2.60782	2.59772	2.96730	2.81379
4	2.85923	2.44282	2.98046	2.68501
5	2.77826	2.68070	2.86739	2.78848
6	2.67859	2.98810	2.94633	2.94590
7	2.70604	2.48703	2.97507	2.88779
8	2.68440	2.54305	3.01596	2.46677
9	2.52757	2.48857	3.01578	2.77184
10	2.63694	2.39842	2.84201	2.63291
11	2.60556	2.63790	3.00695	2.79150
12	2.63597	2.50924	3.03550	2.88082
13	2.43208	2.63417	3.00818	2.88220
14	2.63597	2.51401	2.82852	2.80795
15	2.81549	2.37018	2.96765	2.87176
16	2.63596	2.52585	2.88775	2.89190
17	2.60673	2.29083	2.99339	2.68653
18	2.59561	2.47371	2.94329	2.83522
19	2.39793	2.43440	2.86802	2.65390
20	2.80715	1.97804	2.81148	2.87615
21	2.83780	2.92849	2.86683	2.52029
22	2.64614	2.47472	2.89049	2.75419
23	2.63597	2.72216	3.02900	2.57023
24	2.80468	2.52131	3.02446	2.58864
25	2.54645	2.76343	2.98114	2.63566
26	2.74162	2.67930	2.86908	2.84742
27	2.63187	2.27942	2.84719	2.58816
28	2.63597	2.59817	3.01325	2.62978
29	2.78159	2.82753	2.82101	2.80502
30	2.93888	2.42632	2.95928	2.81514

A brief review of Table 4 shows that BPSO was capable of finding the best result in the majority of cases, in comparison with the other metaheuristics. The second best algorithm was GA, which acquired the best results in fewer runs, and the third-best algorithm was the DBDE with only two hits, whereas

CSA showed the worst performance of all. To statistically validate the mentioned observations, we used the nonparametric Wilcoxon test over the 100 runs of the complete experiment.

In the Wilcoxon paired test, the null hypothesis is that two independent data samples come from continuous distributions with equal medians. In contrast, the alternative hypothesis is that those samples are different in that regard. Usually, either the rejection or the failure to reject the null hypothesis considers a significance level of 5%. The test mentioned above was applied to the experimental data, thus obtaining the results shown in Table 5. After reviewing those values, we can conclude that the best algorithm was BPSO, which was capable of finding better values in the majority of the runs. According to the p -values obtained from the experiment, it is clear that the results acquired by every algorithm came from different medians; therefore, the following arrangement of metaheuristics can be given, from best to worst performance in this task, as BPSO, GA, DBDE, and CSA.

Table 5. p -Values of the Wilcoxon test.

	vs. GA	vs. CSA	vs. DBDE	vs. BPSO
GA	1	0.0017	3.6472×10^{-12}	3.1133×10^{-28}
CSA	0.0017	1	1.9761×10^{-5}	1.7849×10^{-27}
DBDE	3.6472×10^{-12}	1.9761×10^{-5}	1	9.2115×10^{-21}
BPSO	3.1133×10^{-28}	1.7849×10^{-27}	9.2115×10^{-21}	1

The average and standard deviation of the 100 runs are given in Table 6. The information confirms that BPSO was, on average, the best metaheuristic to solve the design of pipe with micro-grooves. Additionally, the algorithm achieved improvements in the capillarity angle of around 11% in comparison with a similar study [21].

Table 6. Average and standard deviation of the best individuals found after 100 runs.

Description	Variable	CSA	GA	BPSO	BDE
Pipe inner radius	R	0.0259 (0.0020)	0.0322 (0.0057)	0.0316 (0.0061)	0.0321 (0.0054)
Contact angle	β	0.4178 (0.2080)	0.5929 (0.3523)	0.4426 (0.3000)	0.6184 (0.3206)
Micro-groove angle	γ	1.0539 (0.2815)	0.9644 (0.3446)	0.9696 (0.3542)	1.0985 (0.2903)
Saturated temperature	st	347.55 (53.47)	414.25 (61.52)	455.33 (23.00)	444.05 (61.81)
Groove depth	p	2.75×10^{-4} (4.79×10^{-5})	3.85×10^{-4} (7.48×10^{-5})	3.85×10^{-4} (8.09×10^{-5})	4.24×10^{-4} (6.69×10^{-5})
Apex angle	ϕ	0.4742 (0.2764)	0.6562 (0.2574)	0.6724 (0.2754)	0.7739 (0.2294)
Equivalent capillary rad	r_m	2.81×10^{-4} (3.51×10^{-5})	3.56×10^{-4} (7.45×10^{-5})	3.59×10^{-4} (7.26×10^{-5})	3.89×10^{-4} (6.99×10^{-5})
Kind of fluid	k_f	1	1	1	1
Type of groove	t_g	0	0	0	0
Capillarity angle	θ^*	2.6430 (0.1578)	2.5718 (0.1886)	2.9229 (0.0931)	2.7424 (0.1236)

4.3. A Time-Based Comparison

The required time in seconds to complete every trial is in Table 7, given in the format mean and standard deviation of the 30 runs. In this experiment, the comparison did not need a statistical validation because the mean values were different for every metaheuristic. In that sense, the best metaheuristic for the micro-groove design was the genetic algorithm (GA), which had the minimum execution time. The second and third best algorithms were discrete binary differential evolution and binary particle swarm optimization, with very similar average times, with a slightly lower standard deviation of the DBDE. In terms of execution time, the clonal selection algorithm had the worst performance, due mainly to the fact that the fitness evaluation of the mutated clones is computationally expensive.

Table 7. Mean and standard deviation of execution times after 30 runs.

CSA	GA	BPSO	DBDE
545.21 (13.59)	38.94 (1.36)	53.19 (2.43)	53.16 (1.84)

5. Conclusions

In this article, were used four metaheuristic algorithms to optimize the pipe design in a parabolic trough: genetic algorithms, binary particle swarm optimization, binary differential evolution, and clonal selection algorithm. In the proposal, we used two differential equations to simulate the dynamic behavior of several fluids contained in a pipe with two kinds of micro-grooves. The experimental results show an averaged improved performance (11%) of the binary particle swarm optimization in comparison with a similar study that explored the design of micro-grooved pipes. After several runs of the algorithm, it is clear that the best tube must have semi-circular micro-grooves, with water as the best working fluid when compared with liquid sodium and two different molten salts. This study also considers the values of angles β and γ to calculate the wet front, producing a better pipe design in a concentrated solar power system than a previous study [21].

Author Contributions: Conceptualization, V.O.-E.; Methodology, M.P.-C.; Software, D.Z.-N.; Validation, V.O.-E., M.P.-C. and D.Z.-N.; Formal Analysis, M.P.-C.; Investigation, V.O.-E.; Resources, M.P.-C.; Data Curation, V.O.-E.; Writing—Original Draft Preparation, V.O.-E.; Writing—Review & Editing, D.Z.-N., M.P.-C.; Visualization, M.P.-C.; Supervision, V.O.-E.; Project Administration, V.O.-E.; Funding Acquisition, M.P.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PRODEP-SEP with the grant number DSA/103.5/15/10814.

Acknowledgments: We appreciate the comments of anonymous reviewers to improve early versions of this document.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Nomenclature

Symbol	Description	Units
A_l	Wetting cross section of the micro-channel	m^2
A_{cont}	Surface in contact with the liquid layer	m^2
R	Pipe inner radius	m
β	Contact angle of the liquid steam solid interface	rad
γ	Micro-channel angle to the vertical	rad
st	Saturated temperature	K
p	Groove's depth in triangular micro-groove	m
ϕ	Apex angle in triangular micro-groove	rad
θ	Angular liquid front position	rad
r_m	Radius of semicircular groove	m
r_H	Hydraulic radius of the micro-channel	m
r_c	Effective of equivalent capillary radius in the direction of interest	m
t	Time	s
ρ_l	Liquid density	kg/m^3
μ_l	Dynamic liquid viscosity	mPa s
σ_l	Surface tension	mN/m
g	Gravitational acceleration	m/s^2
k_f	Kind of fluid: liquid sodium, HITEC salt, molten salt	none
v_l	Liquid front velocity	m/s
tg	Type of groove: triangular, semi-circular	none

Appendix B. Fluid Characteristics

This paper reports the utilization of four fluids of common use in the design of concentrated solar power systems [37–39]. Some useful features of those liquids are in Table A1.

Table A1. Fluids utilized.

<i>kf</i>	Fluid	Chemical Name	Temperature Range (K)
0	Liquid Sodium	Na	371 – 1600
1	Water	H ₂ O	233 – 643
2	HITEC Salt	NaNO ₃ -NaNO ₂ -KNO ₃	450 – 1050
3	Molten Salt	KCl-MgCl	750 – 1550

Appendix C. Calculus of Variables for Liquid Sodium

Equations (A1)–(A3) compute for liquid sodium the saturated liquid density, the surface tension, and the liquid dynamic viscosity, respectively [37]:

$$\rho_l = 219 + 275.32 \times \left(1 - \frac{T}{2503.7}\right) + 511.58 \times \left(1 - \frac{T}{2503.7}\right)^{0.5}, \quad (\text{A1})$$

$$\sigma_l = 240.5 \times \left(1 - \frac{T}{2503.7}\right)^{1.126}, \quad (\text{A2})$$

$$\mu_l = \frac{e^{[-6.4406 - 0.3958 \times \ln(T) + \frac{556.835}{T}]}}{1000}, \quad (\text{A3})$$

where T is the temperature of the fluid in Kelvin.

Appendix D. Calculus of Variables for Water

Equations (A4)–(A6) compute for water the saturated liquid density, the surface tension, and the liquid dynamic viscosity, respectively [38]:

$$\rho_l = \frac{0.14395}{0.0112 \left[1 + \left(1 - \frac{T}{649.727}\right)^{0.05107}\right]}, \quad (\text{A4})$$

$$\sigma_l = \frac{134.15}{1000} \times \left(1 - \frac{T}{647.3}\right)^{\left[1.6146 - 2.035 \times \left(\frac{T}{647.3}\right) + 1.5598 \times \left(\frac{T}{647.3}\right)^2 + 0.0 \times \left(\frac{T}{647.3}\right)^3\right]}, \quad (\text{A5})$$

$$\mu_l = \frac{e^{[-3.7188 + \frac{578.919}{-137.546 + T}]}}{1000}, \quad (\text{A6})$$

where T is the temperature of the fluid in Kelvin.

Appendix E. Calculus of Variables for HITEC Salt

Equations (A7)–(A9) compute for molten salt NaNO₃-NaNO₂-KNO₃ in the ratio 7:49:44 mol% the saturated liquid density, the surface tension, and the liquid dynamic viscosity, respectively [39]:

$$\rho_l = 2293.6 - 0.7497 \times T, \quad (\text{A7})$$

$$\sigma_l = \frac{0.14928 - 0.556 \times 10^{-4} \times T}{1000}, \quad (\text{A8})$$

$$\mu_l = \frac{0.4737 - .002297 \times T + 3.731 \times 10^{-6} \times T^2 - 2.019 \times 10^{-9} \times T^3}{1000}, \quad (\text{A9})$$

where T is the temperature of the fluid in Kelvin.

Appendix F. Calculus of Variables for Molten Salt

Equations (A10)–(A12) compute for the molten salt KCl-MgCl₂, in the ratio 67:33 mol%, the saturated liquid density, the surface tension, and the liquid dynamic viscosity, respectively [39]:

$$\rho_l = 2363.84 - 0.474 \times T, \quad (\text{A10})$$

$$\sigma_l = \frac{0.133 - 0.48 \times 10^{-4} \times T}{1000}, \quad (\text{A11})$$

$$\mu_l = \frac{1.46 \times 10^{-4} \times e^{(2230/T)}}{1000}, \quad (\text{A12})$$

where T is the temperature of the fluid in Kelvin.

Appendix G. Micro-Groove Geometry

The calculus of the variables connected with the micro-groove geometry are in Table A2.

Table A2. Micro-groove parameters (Source: [21]).

tg	Micro-Groove	r_H	r_c	A_l	A_{cont}
0	Semicircular	$\frac{1}{2}r_m$	r_m	$\frac{1}{2}\pi r_m^2$	$\pi r_m R \theta \cos \gamma$
1	Triangular	$\frac{1}{2}p \cdot \sin \frac{\phi}{2}$	$p \cdot \sin \frac{\phi}{2}$	$p^2 \cdot \tan \frac{\phi}{2}$	$\frac{2pR\theta}{\cos \frac{\phi}{2}} \cos \gamma$

References

- Ackermann, T. *Wind Power in Power Systems*; John Wiley: Chichester, UK, 2005; pp. 1–692.
- Chennouf, N.; Negrou, B.; Dokkar, B.; Settou, N. Valuation and estimation of geothermal electricity production using carbon dioxide as working fluid in the south of Algeria. *Energy Procedia* **2013**, *36*, 967–976. [\[CrossRef\]](#)
- Brabec, C.; Sariciftci, N.; Hummelen, J. Plastic solar cells. *Adv. Funct. Mater.* **2001**, *11*, 15–26. [\[CrossRef\]](#)
- Demirbas, A. Progress and recent trends in biofuels. *Prog. Energy Combust. Sci.* **2007**, *33*, 1–18. [\[CrossRef\]](#)
- Hoffert, M.; Caldeira, K.; Benford, G.; Criswell, D.; Green, C.; Herzog, H.; Jain, A.; Kheshgi, H.; Lackner, K.; Lewis, J.; et al. Engineering: Advanced technology paths to global climate stability: Energy for a greenhouse planet. *Science* **2002**, *298*, 981–987. [\[CrossRef\]](#)
- Gupta, K.; Rehman, A.; Sarviya, R. Bio-fuels for the gas turbine: A review. *Renew. Sustain. Energy Rev.* **2010**, *14*, 2946–2955. [\[CrossRef\]](#)
- Pelc, R.; Fujita, R. Renewable energy from the ocean. *Mar. Policy* **2002**, *26*, 471–479. [\[CrossRef\]](#)
- Bajpai, P.; Dash, V. Hybrid renewable energy systems for power generation in stand-alone applications: A review. *Renew. Sustain. Energy Rev.* **2012**, *16*, 2926–2939. [\[CrossRef\]](#)
- Chu, S.; Majumdar, A. Opportunities and challenges for a sustainable energy future. *Nature* **2012**, *488*, 294–303. [\[CrossRef\]](#)
- Odeh, S.; Morrison, G.; Behnia, M. Modelling of parabolic trough direct steam generation solar collectors. *Sol. Energy* **1998**, *62*, 395–406. [\[CrossRef\]](#)
- Shen, Y.; Zhang, H.; Xu, H.; Yu, P.; Bai, T. Maximum heat transfer capacity of high temperature heat pipe with triangular grooved wick. *J. Cent. South Univ.* **2015**, *22*, 386–391. [\[CrossRef\]](#)
- Lu, J.; Ding, J.; Yu, T.; Shen, X. Enhanced heat transfer performances of molten salt receiver with spirally grooved pipe. *Appl. Therm. Eng.* **2015**, *88*, 491–498. [\[CrossRef\]](#)
- Weinstein, L.A.; Loomis, J.; Bhatia, B.; Bierman, D.M.; Wang, E.N.; Chen, G. Concentrating Solar Power. *Chem. Rev.* **2015**, *115*, 12797–12838. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, K.J.; Yang, J.W. Sunlight concentrator design using a revised genetic algorithm. *Renew. Energy* **2014**, *72*, 322–335. [\[CrossRef\]](#)

15. Nazemi, S.; Boroushaki, M. Design, Analysis and Optimization of a Solar Dish/Stirling System. *Int. J. Renew. Energy Dev.* **2016**, *5*, 33–42. [\[CrossRef\]](#)
16. Zhao, J.; Song, Y.; Lam, W.H.; Liu, W.; Liu, Y.; Zhang, Y.; Wang, D. Solar radiation transfer and performance analysis of an optimum photovoltaic/thermal system. *Energy Convers. Manag.* **2011**, *52*, 1343–1353. [\[CrossRef\]](#)
17. Petersont, G.; Lu, X.; Peng, X.; Wang, B. Analytical and experimental investigation of the rewetting of circular channels with internal V-grooves. *Int. J. Heat Mass Transf.* **1992**, *35*, 3085–3093. [\[CrossRef\]](#)
18. Jaramillo, O.; Borunda, M.; Velazquez-Lucho, K.; Robles, M. Parabolic trough solar collector for low enthalpy processes: An analysis of the efficiency enhancement by using twisted tape inserts. *Renew. Energy* **2016**, *93*, 125–141. [\[CrossRef\]](#)
19. Hübner, S.; Eck, M.; Stiller, C.; Seitz, M. Techno-economic heat transfer optimization of large scale latent heat energy storage systems in solar thermal power plants. *Appl. Therm. Eng.* **2016**, *98*, 483–491. [\[CrossRef\]](#)
20. Muñoz, J.; Abánades, A. Analysis of internal helically finned tubes for parabolic trough design by CFD tools. *Appl. Energy* **2011**, *88*, 4139–4149. [\[CrossRef\]](#)
21. Dung, N.D.T.; Wang, K.J.; Chuang, F.S.; Kung, K.Y. Optimizing the design of receiver in parabolic trough by using genetic algorithm. *Eur. J. Mech. B/Fluids* **2015**, *49 Pt A*, 146–152. [\[CrossRef\]](#)
22. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
23. Rojas, M.; de Andrés, M. Theoretical and experimental study of two-phase flow in micro-channels grooved into horizontal pipes. *Int. J. Multiph. Flow* **2006**, *32*, 517–526. [\[CrossRef\]](#)
24. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)
25. Hoffmann, J.F.; Henry, J.F.; Vaitilingom, G.; Olives, R.; Chirtoc, M.; Caron, D.; Py, X. Temperature dependence of thermal conductivity of vegetable oils for use in concentrated solar power plants, measured by 3omega hot wire method. *Int. J. Therm. Sci.* **2016**, *107*, 105–110. [\[CrossRef\]](#)
26. Benoit, H.; Spreafico, L.; Gauthier, D.; Flamant, G. Review of heat transfer fluids in tube-receivers used in concentrating solar thermal systems: Properties and heat transfer coefficients. *Renew. Sustain. Energy Rev.* **2016**, *55*, 298–315. [\[CrossRef\]](#)
27. Dung, N.; Chuang, F.S.; Wang, K.J. Capillary-driven flow analysis of a micro-grooved pipe. *Contin. Mech. Thermodyn.* **2014**, *26*, 423–435. [\[CrossRef\]](#)
28. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
29. Das, S.; Suganthan, P. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [\[CrossRef\]](#)
30. Chen, P.; Li, J.; Liu, Z. Solving 0-1 knapsack problems by a discrete binary version of differential evolution. In Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application, Shanghai, China, 20–22 December 2008; Volume 2, pp. 513–516.
31. Farmer, J.; Packard, N.; Perelson, A. The immune system, adaptation, and machine learning. *Phys. D Nonlinear Phenom.* **1986**, *22*, 187–204. [\[CrossRef\]](#)
32. De Castro, L.; Von Zuben, F. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* **2002**, *6*, 239–251. [\[CrossRef\]](#)
33. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Western Australia, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
34. Khanesar, M.; Teshnehlab, M.; Shoorehdeli, M. A novel binary particle swarm optimization. In Proceedings of the 2007 Mediterranean Conference on Control and Automation, MED, Athens, Greece, 27–29 June 2007; pp. 1–6.
35. Whitley, D. An overview of evolutionary algorithms: Practical issues and common pitfalls. *Inf. Softw. Technol.* **2001**, *43*, 817–831. [\[CrossRef\]](#)
36. Yang, X.S.; Deb, S.; Loomes, M.; Karamanoglu, M. A framework for self-tuning optimization algorithm. *Neural Comput. Appl.* **2013**, *23*, 2051–2057. [\[CrossRef\]](#)
37. Fink, J.; Leibowitz, L. *Thermodynamic and Transport Properties of Sodium Liquid and Vapor*; Argonne National Lab.: Argonn, IL, USA, 1995. [\[CrossRef\]](#)

38. GmbH, D. Saturated Liquid Density. 2016. Available online: <http://ddbonline.ddbst.de/DIPPR105DensityCalculation/DIPPR105CalculationCGI.exe?component=Methanol> (accessed on 1 December 2019).
39. Sohal, M.S.; Ebner, M.A.; Sabharwall, P.; Sharpe, P. *Engineering Database of Liquid Salt Thermophysical and Thermochemical Properties*; Idaho National Laboratory: Idaho Falls, ID, USA, 2010. [CrossRef]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).