Selfish Herd Optimizer for Solving Dynamic Economic Dispatch with Valve-Point Effects

Primitivo Díaz, Adrián Gonzalez, Fernando Fausto, Orlando Salas
Departamento de Electro-Fotónica, Universidad de Guadalajara.
CUCEI, Av. Revolución 1500, Guadalajara Jal. México.

Abstract: **Efficient operation of power generation systems has become a significant issue in the research community, driven by the increasing demand for sustainable energy solutions and mitigating the environmental impact. Dynamic Economic Dispatch (DED) is a well-known optimization problem in power system operation. To accurately model DED, it is crucial to account for valve-point loading effects and ramp-rate limits. These factors make DED a non-linear, non-convex, and non-smooth problem, which is challenging to solve using traditional optimization techniques. In contrast, the Selfish Herd Optimizer (SHO) is a novel algorithm capable of finding optimal solutions to complex problems. In this work the SHO algorithm is used to optimize the DED problem, to evaluate the effectiveness of the proposed approach, the SHO algorithm was applied to three distinct power generation systems. The results obtained demonstrate the competitive performance of the SHO algorithm.**

1. Introduction.

The Economic Dispatch (ED) problem plays an essential role in power system operation. The principal objective of ED is to obtain the minimum operation cost needed to satisfy power balance, generating, and network operating limit constraints [1]. Economic dispatch is a process of efficiently sharing the total load on a power system among various generating plants to satisfy the load demand efficiently. The solution to the ED problem is not a trivial process that requires the use of novel optimization methods capable of solving problems with non-linear, non-convex, and non-smooth characteristics. The principal goal of these optimization techniques in ED is to optimize the energy generated by the system's generating units by minimizing fuel cost while meeting the load demand and power system constraints. An implicit benefit of optimizing power generation units is to assure higher reliability and security of the power system [2].

On the other hand, the dynamic version of the economic dispatch problem aims to manage the generated power output by using schedules within a period of time, considering physical constraints such as power and ramp-rate limits. This approach makes the DED a more complex problem due to the increased dimensionality compared to the ED [3]. Since the DED problem was introduced, several optimization techniques and procedures have emerged in the literature as an alternative to find optimal solutions. The most well-known traditional methods included linear programming [4] and non-linear programming [5], iterative lambda method [6], and Lagrangian relaxation [7], among others. Nevertheless, traditional methods commonly find suboptimal solutions for non-smooth or non-convex problems such as DED with valve-point effects.

The DED has been addressed from alternatives optimization perspectives to overcome these problems. Within metaheuristic techniques, the particle swarm optimization [8], ant colony optimization [9], genetic algorithm [10], differential evolution algorithm [11], and gravitational search algorithm [12] have been proposed. Although these methods achieve promising solutions, they still present certain shortcomings in balancing local exploitation and global exploration. More recent papers implement hybrid algorithms to improve global searching performance and accelerate the

https://doi.org/10.61728/AE20255282

convergence speed. These hybrid algorithms consist of a combination of existing algorithms such as PSO and TCO [3] or a combination of existing algorithms with chaotic searches such as an Improved-PSO [13], Differential bee colony [14], and Differential Evolution methods [15].

The Selfish Herd Optimizer is a novel swarm optimization algorithm that has attracted attention by its remarkable performance [16]. The SHO has been applied to solve diverse engineering problems [17-20]. At the same time, novel variants have emerged to improve its performance while maintaining the essence of the SHO. Among the most notable ones are the Refined Selfish Herd Optimizer (RSHO) [21], a Chaotic Selfish Herd Optimizer (CSHO) [22], and Selfish Herd Optimizer with Levy-Flight [23].

This paper is organized as follows, in section 2 the dynamic economic dispatch problem defined, with the constraints handled, section 3 describes the Selfish Herd Optimizer as well as three improved versions of the same, in section 4 is explained how these algorithms were implemented for DED problem, the results of this work are in section 5, and the conclusions are present in section 6.

2. Problem Formulation

The Dynamic Economic Dispatch is addressed as to simultaneously minimize the power production cost and meet the power system's load demand as well as several other systems constraints over the operating horizon and can be defined as:

$$F = min \sum_{t=1}^{T} \sum_{i=1}^{N} f_i(p_i^t) \tag{1}$$

where F is the total fuel cost over all the dispatch periods. T is the number of intervals in the time horizon. N is the number of generating units. p_i^t is the power output of ith generating unit at time t. $f_i(p_i^t)$ is the fuel cost of ith generating unit at the output of p_i^t.

Usually, the fuel cost of each generating unit is represented as a quadratic convex polynomial function and is defined as:

$$f_i(p_i) = a_i + b_i p_i + c_i p_i^2 \tag{2}$$

where a_i, b_i, and c_i are the coefficients of the generating unit i. Nevertheless, in real power systems with large steams turbines their steam valves produce a considerable increment in fuel cost due to the wire drawing effect each time that the valves are opened which makes the practical objective function (2) have no-differentiable points. Hence, in order to make the cost function of the generation units according to real systems a sinusoidal component is included, as defined on equation (3).

$$f_i(p_i) = a_i + b_i p_i + c_i p_i^2 + \left| d_i sin\left(e_i(p_{i,min} - p_i)\right)\right| \tag{3}$$

where d_i, and e_i are the coefficients of the generating unit i.

2.1. Constraints

i. Power Balance.

The sum of the power generated by each generating unit has to be equal to the power demanded by the load plus the total transmission loss of each time interval *t*.

$$\sum_{i=1}^{N} p_i^t = p_D^t + p_L^t \tag{4}$$

where p_D^t is the total load demand for time interval *t* and p_L^t is the transmission power loss on time interval t, and is calculated with Kron's loss formula known as B-matrix coefficients:

$$p_L^t = \sum_{i=1}^{N}\sum_{j=1}^{N} p_i^t B_{ij} p_j^t + \sum_{j=1}^{N} B_{0i} p_i^t + B_{00} \tag{5}$$

where B_{ij} is the *ij*th element of the loss coefficient square matrix. B_{0i} is the *i*th element of the loss coefficient vector and B_{00} is the loss coefficient constant.

ii. Operating Limits

$$p_{i,min} \leq p_i^t \leq p_{i,max} \quad i = 1,2,\dots N; t = 1,2,\dots T \tag{6}$$

where the $p_{i,min}$ and $p_{i,max}$ are minimum and maximum power output for generation unit *i*.

iii. Ramp rate limits

In order to prolong the life of generators, it is necessary to insure the thermal stress within the acceptable range of turbine [16].

This mechanical restriction is usually converted to a constraint on the rate of increase (ramp-up) or decrease (ramp-down) of the generator output which is called ramp rate limits.[14]. The ramp limits are included in DED problem to ensure the units works on a feasible zone and are defined as:

$$\begin{cases} p_i^t - p_i^{t-1} \leq UR_i \\ p_i^{t-1} - p_i^t \leq DR_i \end{cases} \quad i = 1,2,\dots N; t = 1,2,\dots T \tag{7}$$

where UR_i and DR_i are the up-rate and down-ramp limits of the *i*th generator.

3. Algorithm's overview.

The Selfish Herd Optimizer is a novel swarm optimization algorithm proposed by Fausto et. al (2017) for solving optimization problems [16]. The algorithm is inspired in the behaviors described on Hamilton's selfish herd theory (1971) [19]. The algorithm considers two different kinds of search agents: predators and prey. Each of these agents' movements are conducted by a set of unique rules and operators based on the observed natural behavior of individuals on a selfish herd while they are endangered by a pack of hungry predators. [17].

3.1. Original Selfish Herd Optimizer.

3.1.1.Population Initialization.

The algorithms begins with a set X of individuals that includes two groups of animals, the preys $\boldsymbol{H}$ and the predators $\boldsymbol{P}$, each individual position (X= {x_1, x_2, …, x_N}) where N is the total number of individuals is represented by a n-dimensional vector $\boldsymbol{x}_i$=[$x_{i,1}$, $x_{i,2}$, …, $x_{i,n}$] which will represent a feasible solution for a certain optimization problem. The positions are randomly initialized as follows:

$$an_{i,j} = lim_j^{low} + rand(0,1) * \left(lim_j^{high} - lim_j^{low}\right)$$
$$i = 1,2,\dots N \quad j = 1,2,\dots n \tag{8}$$

where lim_j^{high} and lim_j^{low} represent the upper and lower limits of the search space respectively as well as i and j correspond to the individual index and parameter index of each animal. The *rand*(0,1) denotes a random real number in the interval of [0,1].

3.1.2.Population split.

SHO employs two different types of search agents: the herd H={h_1,h_2,…h_{Nh}} and the predators P={p_1,p_2,…p_{Np}},where N_h corresponds to the total number of preys H and N_p corresponds to the total number of predators P, such that A=H∪P and the number of individuals for each group is randomly selected as follows:

$$N_h = floor\left(N * rand(0.7,0.9)\right) \tag{9}$$

$$N_p = N - N_h \tag{10}$$

where *rand(0.7,0.9)* denotes a random number between [0.7,0.9] and *floor()* maps areal number to an integer number.

3.1.3.Survival value.

In SHO the survival capability of each individual a_i (either the chance of survive a predator's attack or the chance to success killing a prey accordingly) is assigned with a survival value SVa_i and it represents its survival aptitude which is the quality of the solution and is relative to the current position within the solution space. The SV_{ai} is defined as:

$$SV_{ai} = \frac{f(a_i) - f_{best}}{f_{best} - f_{worst}} \tag{11}$$

where $f(a_i)$ denotes the fitness value corresponding to the evaluation of the objective function *f()* for individual ai. f_{best} and f_{worst} corresponds to the best and worst fitness values found so far by the evolutionary process of the algorithm. Considering a maximization problem these values are defined as:

$$f_{best} = max\left(\left(max(f(a_i))\right)_j\right) \tag{12}$$

$$f_{worst} = min\left(\left(min(f(a_i))\right)_j\right) \tag{13}$$

$$j \in \{0,1,2,\dots k\};\ i \in \{0,1,2,\dots N\}$$

where k is the current iteration

3.1.4.Selfish Herd movements.

SHO divides the herd population in three different roles in order to model the distinctive decision-making behaviors.

3.1.4.1. Herd's leader.

The leader of the herd is the individual whose position inside the herd aggregation promotes it to have the highest chances of surviving a predador attack [Eshel et al. 2011]. Hence at each iteration k, the algorithm assigns a single individual h_i as the leader of the selfish herd. The individual h_L is chosen by considering the current survival value of each individual of the herd's aggregation as:

$$h_L^k = \left(h_i^k \in H^k \left| SV_{h_i^k} = \underset{j \in \{1,2,\dots N_h\}}{max}\left(SV_{h_j^k}\right)\right.\right) \tag{14}$$

The prey individual possessing the highest survival value among all other members of H is assigned as the leader, this will occur for each iteration.
Then two types of movements are performed based on the survival value of the leader, these movements are *seemingly cooperative leadership* and *openly selfish leadership*, and they are defined as:

$$h_L^{k+1} = \begin{cases} h_L^k + c^k & if\ SV_{h_L^k} = 1 \\ h_L^k + s^k & if\ SV_{h_L^k} < 1 \end{cases} \tag{15}$$

where $\boldsymbol{c}$ is a movement vector, which represents the decision taken by the leader to guide the herd to a better position and guarantee its safety, this decision will be taken only if the leader has the best position f_{best}. Movement vector $\boldsymbol{c}$ is calculated as:

$$c^k = 2 * \alpha * \left(-SV_{pM}\right) * e^{-\|p_M - h_L\|^2} * (p_M - h_L) \tag{16}$$

where $\left(-SV_{pM}\right)$ represents the survival value related to the predator's center mass as defined by (18). $\boldsymbol{s}$ is also a movement vector and represent the location that will increase the survival value of the leader only, and is defined as:

$$s^k = 2 * \alpha * e^{-\|x_{best} - h_L\|_2^2} * (x_{best} - h_L) \tag{17}$$

where x_{best} represents the best position so far and α is a random number within [0,1]. ‖ ‖ is the Euclidean distance between the positions mentioned. SV_{pM} denotes the survival value related to the predator's center of mass which can be defined as:

$$p_M^k = \frac{\sum_{j=1}^{Np} SV_{p_i}^k * p_i^k}{\sum_{j=1}^{Np} SV_{p_j}^k} \tag{18}$$

where SV_{pj} is the survival value of the individual p_j. The predator's center of mass is how SHO calculates a position whit relatively higher risk for an individual of the herd. In same order of ideas SHO considers the existence of relatively safer central positions within the herd and is given by the herd's population center of mass defined as:

$$h_M^k = \frac{\sum_{j=1}^{Nh} SV_{h_i}^k * h_i^k}{\sum_{j=1}^{Nh} SV_{h_j}^k} \tag{19}$$

Since both h_M and p_M represents potential solutions within the solution space of a given optimization problem, a corresponding survival value can be computed for each position with (11).

3.1.4.2. Nearest best neighbors.

The algorithm considers the nearest best neighbor for any individual hi is whose possess two important traits: 1) It is the nearest herd member other than the leader h_L. 2) It has better aptitudes than the individual hi. The nearest best neighbor is defined as:

$$h_{ci}^k = \left(h_j^k \in H^k, h_j^k \neq \left[h_j^k, h_L^k\right] \middle| SV_{h_j^k} > SV_{h_i^k}, r_{ij} = \min_{j \in \{1,2,\dots N_h\}} \left(\left\|h_i^k - h_j^k\right\|\right)\right) \tag{20}$$

where r_{ij} is the Euclidean distance between indexed herd members i and j.

3.1.4.3. Herd's followers and deserters.

SHO classifies the members of the aggregation in two different groups, the herd followers H_F and herd's deserted H_D and the movement pattern performed by the herd depends on the role assumed by each individual. These two groups are defined as:

$$H_F^k = \left\{h_j^k \neq h_L^k \middle| SV_j^k \geq rand(0,1)\right\} \tag{21}$$

$$H_D^k = \left\{h_j^k \neq h_L^k \middle| SV_j^k < rand(0,1)\right\} \tag{22}$$

The position update of each individual is defined at each iteration by:

$$h_i^{k+1} = \begin{cases} h_i^k + f_i^k \;\; if \;\; h_i^k \in H_F^k \\ h_i^k + f_i^k \;\; if \;\; h_i^k \in H_D^k \end{cases} \tag{23}$$

where f_i^k is a movement vector computed with regard to the positions and survival aptitudes of other members of the herd. In addition, the members of the herd followers H_F are subdivided in two groups, the dominant H_d and the subordinates H_s, depending on their survival values in relation with the mean survival value of the herd population. Therefore, the movement vector f_i^k performed by each member of the group H_F^k is calculated depending on whether it is a dominant or subordinated member:

$$f_i^k = \begin{cases} 2 * \left(\beta * \psi_{h_i,h_L}^k * \left(h_L^k - h_i^k\right) + \gamma * \psi_{h_i,h_{ci}}^k * \left(h_{ci}^k - h_i^k\right)\right) & if \; h_i^k \in H_d^k \\ 2 * \delta * \psi_{h_i,h_M}^k * \left(h_M^k - h_i^k\right) & if \;\; h_i^k \in H_s^k \end{cases} \tag{24}$$

where β, γ and δ are random numbers in the interval [0,1]. And the subgroups H_d and H_s are defined in equations (25) and (26). Same case for ψ_{h_i,h_L}^k and $\psi_{h_i,h_{ci}}^k$ which are the selfish attraction experimented by herd member h_i toward the leader h_L and its best nearest neighbor h_{ci} as defined in equations (27) and (28). While ψ_{h_i,h_M}^k represent the attraction experimented by h_i towards the center of mass h_M as defined in equation (29)

$$H_d^k = \left\{h_i^k \in H_F^k \middle| SV_{h_i}^k \geq SV_{H_\mu}^k\right\} \tag{25}$$

$$H_s^k = \left\{h_i^k \in H_F^k \middle| SV_{h_i}^k < SV_{H_\mu}^k\right\} \tag{26}$$

where $SV_{H_\mu}^k$ is the mean survival value of H at iteration k and is defined by:

$$SV_{H_\mu}^k = \frac{\sum_{i=1}^{N_h} SV_{h_i}^k}{N_h^k} \tag{27}$$

The attraction experimented by a herd member towards its leader and its nearest best member is defined as:

$$\psi_{h_i,h_L}^k = SV_{h_L} * e^{-\|h_i - h_L\|^2} \tag{28}$$

$$\psi_{h_i,h_{ci}}^k = SV_{h_{ci}} * e^{-\|h_i - h_{ci}\|^2} \tag{29}$$

$$\psi_{h_i,h_M}^k = SV_{h_M} * e^{-\|h_i - h_M\|^2} \tag{30}$$

In the case of the deserters H_d moves in a different way, based on desertion rule, independently of any other individual:

$$d_i^k = 2 * \left(\beta * \psi_{h_i,x_{best}}^k * \left(x_{best}^k - h_i^k\right) + \gamma * \left(1 - SV_{h_i^k}\right)\hat{r}\right) \tag{31}$$

where $\psi_{h_i,x_{best}}^k$ denotes is the selfish attraction experienced by the individual h_i^k toward the current best position and is defined in (32), β, and γ represents random numbers from the interval [0,1], also $\hat{r}$ is a unit vector pointing to a random direction within the n-dimensional search space.

$$\psi_{h_i,x_{best}}^k = e^{-\|h_i - h_{best}\|^2} \tag{32}$$

3.1.5.Predators' movement.

The first step to model the movement of the predators is to assume that each herd member has a certain probability of being pursued by an attacking predator, this probability is computed in SHO as:

$$\mathcal{P}_{p_i,h_j} = \frac{\omega_{p_i,h_j}}{\sum_{m=1}^{N_h} \omega_{p_i,h_m}} \tag{33}$$

where ω_{p_i,h_j} represents the prey attractiveness between p_i and h_j, this value considers consider the survival value of the herd member hi as well as the distance between the prey and the attacking predator p_j as defined:

$$\omega_{p_i,h_j} = \left(1 - SV_{h_j}\right) * e^{-\|p_i - h_j\|^2} \tag{34}$$

The prey attractiveness ω_{p_i,h_j} yield to higher values toward to herd members possessing lower survival values, whereas a higher survival value implies lower attractiveness value, this is analogous to the preference of a predator to attack weaker preys. Similarly, the value ω_{p_i,h_j} will be increased as the distance between the predator and the prey $\|p_i - h_j\|$ decreases, and vice versa. For the predators position update is considered the position of particular individual of the prey herd as defined below:

$$p_i^{k+1} = p_i^k + 2 * \rho * \left(h_r^k - p_i^k\right) \tag{35}$$

where ρ is a random number in the range of [0,1]. Furthermore $h_r^k = h_j^k \in H^k \quad r \in \{1,2,\dots N_h\}$ represent a herd member randomly chosen from the whole members of the herd $H^k$by applying the roulette selection method (Thomas,1996) [20] with regard to their individual pursuit probabilities $\mathcal{P}_{p_i,h_j}$ defined in (33).

3.1.6.Recalculation of survival values.

After the herd's and predator's movements the survival value of the entire population could change, with this in mind, the survival value of all individuals is recalculated with equation (11).

3.1.7.Predation phase.

In SHO the predation phase is implemented to model the biological interaction between groups of prey and predators in which former individuals are hunted by the latter, and possible killed and later consumed. It is assumed that after both individuals, the prey and the predator have performed a movement, there is a chance for several herd members to be killed by attacking predators, which implies the exclusion of their respective solution from the search space. This process starts with the implementation of domain of danger which refers to the area where the prey is under a higher risk to be attacked, and it is a circular area around each predator member defined as:

$$R = \frac{\sum_{j=1}^{n} \left| lim^{low} - lim^{high} \right|}{2 * n} \tag{36}$$

Where lim^{low} and lim^{high} represent the lower and upper limits for the search space, and n is the number of dimensions, for simplicity SHO assumes that the radius of the domain of danger is the same for all preys.

Starting the predation phase, it is assumed that no prey has been killed, in order to keep a track on the number of preys killed, SHO initializes an empty set K that during the predation phase is used to group all prey members killed by a predator:

$$K = \{0\} \tag{37}$$

A predator will be able to success in its attack only if two specific condition are met:1) the prey *hj* has a lower survival value than pi and 2) the distance between pi and h_j is equal or lower than the domain of danger radius *R* as defined in equation (36) which implies that each predator is able to threat more than one prey individual, if and only if the two conditions are met. This can be defined as:

$$T_{p_i} = \left\{ h_j \in H \left| SV_{h_j} < SV_{p_i}, \left\| p_i - h_j \right\| \leq R, h_j \notin K \right. \right\} \tag{38}$$

where $\left\| p_i - h_j \right\|$ is the Euclidian distance between p_i and h_j, and T_{p_i} is a set of preys threatened by predator pi, is important to mention that the preys at this step are only threatened by one predator

only, if any of these prey members is killed, is randomly chosen by the roulette selection based on the probabilities to be hunted and it can be defend as:

$$\mathcal{H}_{p_i,h_j} = \frac{\omega_{p_i,h_j}}{\sum_{\left(h_m \in T_{p_i}\right)} \omega_{p_i,h_m}}, h_j \in T_{p_i} \tag{39}$$

where ω_{p_i,h_j} represents the prey attractiveness between the prey and predator as defined in equation (34).

3.1.8.Restoration phase.

In this step, the herd members that were killed during predation phase and stored in K are replaced implementing a special mating-like operator to generate new solutions based on the survival aptitudes of the remaining herd members. This operator considers positions and survival values of all survivors herd members, these survivors can be considered as a set of mating candidates M as defined below:

$$M = \{h_j \notin K\} \tag{40}$$

Each member h_j is considered to have a certain probability $\mathcal{M}_{h_j}$ of being taking into account for the generation of new solution, and such probability is defined as:

$$\mathcal{M}_{h_j} = \frac{SV_{h_j}}{\sum_{(h_m \in M)} SV_{hm}}, h_j \in M \tag{41}$$

As it can be seen, mating candidates possessing higher survival values will have a higher chance to influence the generation of new solutions whereas individuals with a lower survival value are less likely to be considered for such process.

In order to generate a new individual and candidate solution it is considered a set of n randomly chosen individuals selected by the roulette selection method [19] with regard the probabilities $\mathcal{M}_{h_j}$ of each member within the set of mating candidates M as:

$$h_j = h_{new} = mix\left(\left[h_{r_1,1}, h_{r_2,2}, \dots h_{r_i,l}, \dots h_{r_n,n}\right]\right), h_i \in K \tag{42}$$

where n is the dimensionality of the search space and *mix()* is a function that generates a new solution h_{new} by setting the *l*th element $h_{r_i,l}$ of a candidate herd member h_{r_1} to the *l*th component of the h_{new}.

3.2. Refined Selfish Herd Optimizer.

Although the selfish herd optimizer can provide a good performance when finding an optimal solution, it has some oversights that cause problems in its exploitation ability. Additionally, it has shortcomings that influence the exploration and avoiding stagnation in local optima. In SHO the survival value is not only related to the survival aptitude of the individual but also to the decisions made by the predators to pursue specific prey and hence to the risk faced by the herd members. A higher survival value possessed denotes a better solution for solving the problem, the survival value is calculated from the current best and worst fitness values found on each iteration as defined in

equation (11), from this, is easily observed that its value becomes smaller when its corresponding fitness value $f(a_i)$ approaches the best fitness value, and it increases when $f(a_i)$ gets close to the worst fitness value. This might be an oversight not corrected in the literature, an editing mistake [20] and the corrected value is defined as:

$$SV_{ai} = 1 - \frac{f_{best} - f(a_i)}{f_{best} - f_{worst}} \tag{43}$$

This definition avoids the implementation of any programming flags or change the position of the denominator variables according to the application of the algorithm, either for maximization or minimization.

The recalculation of the survival value by SHO is conducted after both, the prey and predator movements are completed. In consequence the survival values of the preys, used in the predator movement phase are not up to date as they should be due to the positions of the herd members change during the herd movement phase. The implementation of these values during the predator moving phase may lead to inappropriate exploitation and exploration of solutions conducted by the predators. This could influence the global convergence performance of SHO. From equation (34) the prey attractiveness is related to the survival value of the preys, using prey attractiveness computed from values that are not updated might result in a higher probability of being pursued for a herd member even when it has increased its survival value in a previous movement phase, this may mislead a predator to an incorrect position. A similar problem occurs in the herd movement in the second iteration when a survival value which has not been updated after a new solution generated in restoration phase is used. Hence the recalculation of the survival values should be done after every movement operator is implemented, either herd movement, predator movement or restoration phase if any new solution has been created.

The domain of danger is implemented as a pair condition, to avoid an imbalance between exploration and exploitation hence it can be considered a range of sub-search area in the entire solution space, in equation (36) after the values of lim^{low} and lim^{high} along with the number of dimensions n are determined, the radius of the domain of danger becomes a constant value during the entire process which means that using a large number of R during all iterations could lose the its functionality during the last phase of the iteration, therefore the equation (41) is suggested to be implemented instead of (36).

$$R = \lambda * n * \|h_{max} - h_{min}\|_1 \tag{44}$$

where λ is a random number from the range [0,1], n is the number of dimensions of the search space, $\| \ \|$ represents the l_1-norm and h_{max} and h_{min} are vectors composed of the largest and smallest components in each dimension of H.

Different types of attractions are used to manifest the behaviors observed from the interaction between preys and predators, these attractions are modeled by using the common factor of the negative natural exponential function, where the square of the Euclidean distance between different vectors is used as input. When only the factor of the exponential is taking into consideration, the movement of the individuals will depend on how close or how apart are these individuals, if a predator is close to a

prey or center of herd, then the predator moves are close to the target, if the predator is far from the target then its movements will be small, in addition due to the squaring, the value computed by this factor is penalized heavily. Thus, even if the distance is not too far from the movement will become extremely small and the predator will remain in the area around its original position (this apply also for preys' movement) the below equations are proposed replace equations (16), (17), (28) to (30) and (32) respectively:

$$c^k = 2 * \alpha * \left(-SV_{pM}\right) * e^{-\|p_M - h_L\|} * (p_M - h_L) \tag{45}$$

$$s^k = 2 * \alpha * e^{-\|x_{best} - h_L\|} * (x_{best} - h_L) \tag{46}$$

$$\psi^k_{h_i,h_L} = SV_{h_L} * e^{-\frac{\|h_i - h_L\|}{n}} \tag{47}$$

$$\psi^k_{h_i,h_{ci}} = SV_{h_{ci}} * e^{-\frac{\|h_i - h_{ci}\|}{n}} \tag{48}$$

$$\psi^k_{h_i,h_M} = SV_{h_M} * e^{-\frac{\|h_i - h_M\|}{n}} \tag{49}$$

$$\psi^k_{h_i,x_{best}} = e^{-\frac{\|h_i - x_{best}\|}{n}} \tag{50}$$

From above equation, n represents the number of dimensions of the search space, and || || the Euclidean distance, x_{best} is the best fitness found so far.

3.3. Chaotic Selfish Herd Optimizer.

It has been proved that SHO can provide competitive results in comparison to other well-known metaheuristics algorithms, and its main problem as in many other algorithms, is that easily get trapped into local solutions with low precision and slow convergence speeds, in order to improve SHO's performance, chaotic search have been added to searching process of SHO [21].

In SHO, a random variable is used to decide whether the herd members are followers or deserters as defined in equation (24) and (31). This parameter plays an important role balancing the exploitation and the exploration of the algorithm because it directly influences the position update of every search agent on every phase.

In this version of SHO, a chaotic sequence generated from a chaotic map is implemented instead of a random variable to improve the performance of the algorithm as defined below:

- If $SV_i \geq CV_k$
  - Apply herd following movement operator as in equation (24)
- Else:
  - Apply herd desertion movement operator as in equation (31)

3.3.1. Chaotic maps

Chaos can be defined as deterministic and arbitrary strategy observed in a dynamic non-linear system which is bounded and non-converging, is the arbitrariness of a basic dynamic deterministic framework and the chaotic system might be considered as a source of randomness [22]. The implementation of these chaotic variables instead of random variables may perform downright

searches at higher speeds in comparison to randomized searches which depends on probabilities [23]. A large number of chaotic maps are available in the field of optimization [24-25]. In this particular application of the algorithm the Tent map was implemented because it improves the exploration of SHO and provides a higher variation in the possible solutions, the Tent map is defined as:

$$x_{k+1} = \begin{cases} 2_{x_k} & x_k < 0.5 \\ 2(1 - x_k) & x_k \geq 0.5 \end{cases} \tag{51}$$

Chaotic sequences are also generated in range [0,1]. For further analysis of all the chaotic maps implemented with SHO, please refer to [21]

3.4. Selfish Herd Optimizer with Levy-Flight

Selfish herd optimizer is a new optimization algorithm; however, its optimization performance is not satisfactory due to the weak global search ability, in order to increase it has been added the Levy flight distribution strategy. [26].

Some theories have proved that Levy-flight distribution strategy exist on the hunt process of the wild animals [27-33]. Based on this, the distribution is applied to the formula for update the position of the herd leader because it is the position of the prey leader the one that represents the global search process, hence with this improvement it will have a better opportunity to find global optimal solutions. The formula proposed is defined as:

$$h_L^{k+1} = h_L^k + \alpha \oplus Levy(\beta) \tag{52}$$

where α is step-sized and is a random number of dimensions, which can be represent as $random(size(n))$, β represents a random number between [0,1], the symbol $\oplus$ represents entry-wise multiplication.

According to (30), the generation of step-sized is defined as follows:

$$g = random(size(n)) \oplus Levy(\beta) \sim 0.01 \frac{\mathcal{L}}{|\nu|^{1/\beta}} * \left(h_j^k - x_{best}^k\right) \tag{53}$$

where ν and $\mathcal{L}$ represents normal distributions and defined as:

$$\nu \sim N(0, \sigma_u^2), \mathcal{L} \sim N(0, \sigma_u^2) \tag{54}$$

having:

$$\sigma_u = \left\{ \frac{\Gamma * (1 + \beta) * sin\left(\frac{\pi * \beta}{2}\right)}{\Gamma * \left(\frac{1 + \beta}{2}\right) * \beta * 2^{(\beta - 1)/2}} \right\}^{1/\beta} \tag{55}$$

where Γ is the standard gamma-function.

The condition for leader's position update is defined as:

$$h_L^{k+1} = \begin{cases} h_L^k + g \quad if \quad SV_{h_L^k} = 1 \; and \; \vartheta < 0.5 \\ h_L^k + c^k \quad if \quad SV_{h_L^k} = 1 \; and \; \vartheta \geq 0.5 \end{cases} \tag{56}$$

where ϑ is a random number from the interval [0.1].

4. SHOs for DED Problem

As the reader can see, the differences between the original SHO and the new versions of SHO are related to the mathematics of the movement operators, or with the decision making of the leader of the herd and the aggregation for their movements. With that being said, the implementation of the algorithms to DED problem will remain essentially the same for all cases, with the few exceptions mentioned accordingly.

- **Step 1:** The initialization of the algorithms, remains the same for all of them, as is defined as in equation (8) with the limits of the space search defined by the maximum and minimum limits of power generation, and ensure that the individuals generated comply with the limits constraint as defined in equation (6)
- **Step 2:** Split the population of set *A* into herd and predators, as defined in equations (9) and (10).
- **Step 3:** Compute the survival values for the population with equation (11), for the case of the Refined SHO, the survival value is computed with equation (43) instead, the cost function defined in equation (3) remains the same for both cases, in addition the power losses for each individual, defined in equation (5).
- **Step 4:** Selfish herd movement operation in this step, for the Chaotic SHO, the herd movement will depend on the value of the chaotic variables, and they are defined as in equation (51). For the SHO with Levy-Flight is in this step when the operator is implemented as defined in equation (52)
- **Step 5a:** Predators movement operation, for the original SHO, the chaotic SHO and LFSHO
- **Step 5b:** The refined SHO will calculate the survival values of the herd with equation (43) after the movement of the herd.
- **Step 6a:** Recalculate the survival values for the entire population, or set A with equation (11) for the original SHO, the chaotic SHO and LFSHO after the completion of steps 4 and 5a
- **Step 6b:** After the completion of the step 5b, the refined SHO will then execute the predator's movements operator.
- **Step 7a:** Predation phase
- **Step 7b:** The refined SHO will perform a Re calculation of the survival values of the predators with equation (43)
- **Step 8a:** Restoration phase
- **Step 8b:** After computes the survival values for refined SHO, the next two steps are the same **Predation** and **Restoration** phase that for the other algorithms.
- **Step 9:** Start the next iteration, *k= k+1* if applicable
- **Step 10:** Output the results, and start next dispatch

The definition of the best and worst positions (f_{best} and f_{worst}) for this specific case, was implemented in a different way than the normal application to find the minimum or maximum value, this because the equality constraint of the DED problem, the best position will be the closest to the power load of the current dispatch.

5. Simulation and Results

In order to review the effectiveness and make a full comparison of the proposed SHO algorithms, these algorithms were implemented in three test cases where the time horizon is split in 24 dispatches for all three cases as defined below:

1. *Case 1*: 5-unit system with valve points effects, ramp-rate limits and considering power transmission losses.
2. *Case 2*: 10-unit system with valve points effects, ramp-rate limits and considering power transmission losses.
3. *Case 3*: 10-unit system with valve points effects and ramp-rate limits without considering power transmission losses.

These three cases are the most used test cases in the literature. The algorithms were implemented in Matlab® 9.9.0 (2020b) and executed on a processor AMD Ryzen™ 5 2500U at 2.00GHz with 8GB on RAM. The number of total individuals A was set to 30 accordingly to [21] for global optimization problems, and 750 iterations to reach the local optima but the Levy-Flight SHO which reaches the local optima with lesser iterations that the other algorithms.

The test was executed 30 times independently for each version of SHO to obtain the results.

Test case 1: 5-unit system with power transmission loss.

For this test case, the valve point effects, the ramp-rate limits and the power transmission losses are considered. The input data for the 5-unit system is displayed on table 1, the loss coefficients matrix implemented to compute the power transmission losses in the 5-unit system is enlisted in table 2, the load demand for 24 hours is enlisted in table 3 and finally figure 1 displays the average total dispatch cost of each SHO version used.

Dispatch	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Power Load (MW)	410	435	475	530	558	608	626	654	690	704	720	740
Dispatch	13	14	15	16	17	18	19	20	21	22	23	24
Power Load (MW)	704	690	654	580	558	608	654	704	680	605	527	463

Table 1.- Input data for test case 1.

4.90E-05	**1.40E-05**	**1.50E-05**	**1.50E-05**	**2.00E-05**
1.40E-05	**4.50E-05**	**1.60E-05**	**2.00E-05**	**1.80E-05**
1.50E-05	**1.60E-05**	**3.90E-05**	**1.00E-05**	**1.20E-05**
1.50E-05	**2.00E-05**	**1.00E-05**	**4.00E-05**	**1.40E-05**
2.00E-05	**1.80E-05**	**1.20E-05**	**1.40E-05**	**3.50E-05**

Table 2.- B-matrix coefficients for test case 1.

Generator	1	2	3	4	5
$p_{i,max}(MW)$	75	125	175	250	300
$p_{i,min}(MW)$	10	20	30	40	50
$a_i\left(\$/_h\right)$	25	60	100	120	40
$b_i\left(\$/_{MW\,h}\right)$	2	1.8	2.1	2	1.8
$c_i\left(\$/_{MW^2\,h}\right)$	0.0080	0.0030	0.0012	0.0010	0.0015
$d_i\left(\$/_h\right)$	100	140	160	180	200
$e_i\left(rad/_{MW}\right)$	0.042	0.040	0.038	0.037	0.035
$UR_i(MW)$	30	30	40	50	50
$DR_i(MW)$	30	30	40	50	50

Table 3.- 24 hours Load demand for test case 1

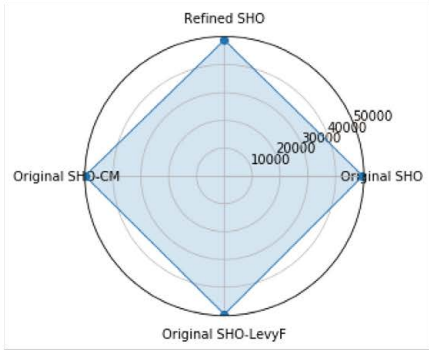


Fig. 1.- Average total cost ($) for test case 1

Table 4 shows a comparison of fuel cost obtained by each version used of SHO against two other algorithms used for the same test case as reported in the literature.

Method	Minimum Cost ($)	Average Cost ($)	Maximum Cost ($)	Transmission loss	Time in secs
CMAES	$43,526.00	$43,915.00	$44,191.00	-	16.36
HPSTCO	-	$42,151.33	-	194.318	58.8
Original SHO	$48,745.52	$49,776.67	$50,987.37	191.433	17.87
Refined SHO	$47,769.13	$48,867.63	$49,656.32	187.904	19.99
CM SHO	$48,682.96	$50,097.77	$51,832.72	189.910	18.78
LF SHO	$47,820.74	$49,767.04	$51,012.32	191.164	0.72

"-" means that the data is not available in the literature.

Table 4.- Algorithms comparison for test case 1

Figueres 2 and 3 displays the convergence time of each SHO algorithm used, a detailed table enlisted the results obtained for each algorithm is present in the appendix section, where table A1 enlists the results of the Original SHO for 5 generation unit system, table A2 enlists the results of the Refined SHO, table A3 enlists the results of Chaotic SHO and table A4 enlists the results of Levy-Flight SHO, figures AF2 trough AF5 displays a graphical representation of these tables.

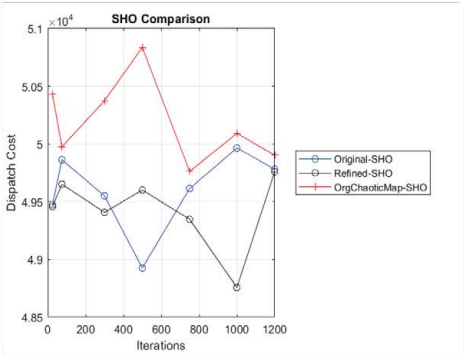


Fig. 2.- Average cost through iterations for test case 1

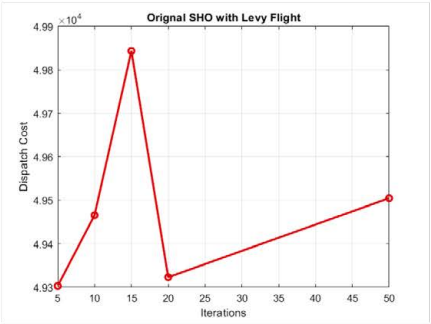


Fig. 3.- Average cost through iterations of the original SHO with Levy-Flight for test case 1

Test case 2: 10-unit system with power transmission loss.

For this test case, the valve point effects, the ramp-rate limits and the power transmission losses are considered. The input data for the 10-unit system is displayed on table 5, the loss coefficients matrix implemented to compute the power transmission losses in the 10-unit system is enlisted in table 6, the load demand for 24 hours is enlisted in table 7 and finally figure 4 displays the average total dispatch cost of each SHO version used, from which the Refined SHO is lowest result obtained. The average total cost which is the sum of the total cost for each dispatch taking into consideration the same generators unit's constraints which are the valve-points effects, the ramp-rate limits, and the power loss, in the literature is in the neighborhood of $1,050,000

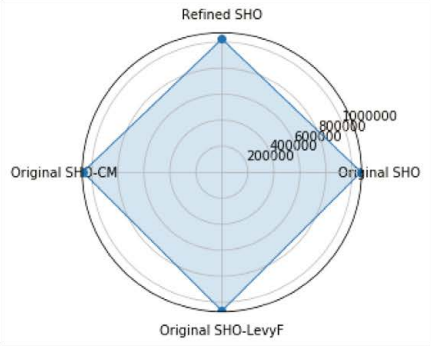


Fig. 4.- Average total cost ($) of test case 2.

The results obtained of each version of SHO are directly compared to the results obtained of the aforementioned hybrid algorithms CDBCO[21] and HPSTCO[3] as well with other two well-known algorithms CMAES[34] and GSA[35]. From Table 8 it is observed that the minimum cost is $953,726.19 computed with the Refined SHO while the fastest algorithm is the LFSHO with an average time of 1.2 sec. this because the number of iterations. In the Appendix section are the results of each algorithm, the table A5 displays the result of the test case for the original SHO, the table A6 displays the results of the refined SHO while table A7 displays the results of the chaotic SHO and table A8 displays SHO with Levy-Flight operator.

Generator	1	2	3	4	5	6	7	8	9	10
$p_{i,max}(MW)$	470	460	340	300	243	160	130	120	80	55
$p_{i,min}(MW)$	150	135	73	60	73	57	20	47	20	55
$a_i\left(^{\$}/_{h}\right)$	958.2	1313.6	604.97	471.6	480.29	601.75	502.7	639.4	455.6	692.4
$b_i\left(^{\$}/_{MW\ h}\right)$	21.6	21.05	20.81	23.9	21.62	17.87	16.51	23.23	19.58	22.54
$c_i\left(^{\$}/_{MW^2\ h}\right)$	0.00043	0.00063	0.00039	0.0007	0.00079	0.00056	0.00211	0.0048	0.10908	0.00951
$d_i\left(^{\$}/_{h}\right)$	450	600	320	260	280	310	300	340	270	380
$e_i\left(^{rad}/_{MW}\right)$	0.041	0.036	0.028	0.052	0.063	0.048	0.086	0.082	0.098	0.094
$UR_i(MW)$	80	80	80	50	50	50	30	30	30	30
$DR_i(MW)$	80	80	80	50	50	50	30	30	30	30

Table 5.- Inputs for test case 2.

The random behavior of the algorithms trough the iterations are represented with figure 5, which compares the total fuel cost obtained by all the algorithms that were executed with the same number of iterations, the Original SHO, Refined and with Chaotic maps varies in the same range, but they never reach stability and figure 6 displays the behavior of LFSHO which indeed reaches the stability.

8.70	**0.43**	**-4.61**	**0.36**	**0.32**	**-0.66**	**0.96**	**-1.60**	**0.80**	**-0.10**
0.43	**8.30**	**-0.97**	**0.22**	**0.75**	**-0.28**	**5.04**	**1.70**	**0.54**	**7.20**
-4.61	**-0.97**	**9.00**	**-2.00**	**0.63**	**3.00**	**1.70**	**-4.30**	**3.10**	**2.00**
0.36	**0.75**	**0.63**	**0.47**	**8.6**	**-0.80**	**0.37**	**0.72**	**-0.90**	**-0.69**
-0.66	**-0.28**	**3.00**	**2.62**	**-0.8**	**11.80**	**-4.90**	**0.30**	**3.00**	**-3.00**
0.96	**5.04**	**1.70**	**-1.96**	**0.37**	**-4.90**	**8.24**	**-0.9**	**5.90**	**-0.60**
-1.60	**1.70**	**-4.30**	**2.10**	**0.72**	**0.30**	**-0.9**	**1.20**	**-0.96**	**0.56**
0.80	**0.54**	**3.10**	**0.67**	**-0.90**	**3.00**	**5.90**	**-0.96**	**0.93**	**-0.30**
-0.10	**7.20**	**-2.00**	**1.80**	**0.69**	**-3.00**	**-0.60**	**0.56**	**-0.30**	**0.99**

*The coefficients are in the order of 10_{π}^{-5}

Table 6.- B-matrix coefficients for test case 2.

Dispatch	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Power Load (MW)	1036	1110	1258	1406	1480	1628	1702	1776	1924	2072	2146	2220
Dispatch	13	14	15	16	17	18	19	20	21	22	23	24
Power Load (MW)	2072	1924	1776	1554	1480	1628	1776	2072	1924	1628	1332	1184

Table 7.- Load demand for 24 hours.

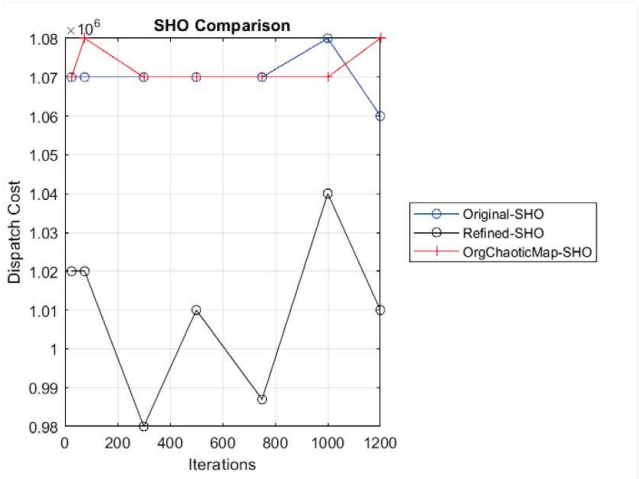


Fig. 5.- Average cost through iterations for test case 2.

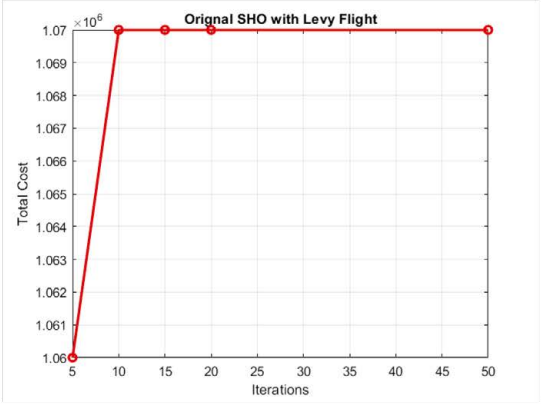


Fig. 6.- Average cost through iterations of the original SHO with Levy-Flight for test case 2.

Method	Minimum Cost ($)	Average Cost ($)	Maximum Cost ($)	Time in secs
GSA	$1,042,964.95	$1,046,216.74	$1,047,362.92	62.37
CDBCO	$1,042,900.00	$1,044,700.00	-	91.8
HPSTCO	$1,035,730.20	$1,036,236.63	$1,036,987.34	111
Original SHO	$1,061,714.19	$1,071,422.82	$1,084,631.13	25.155
Refined SHO	$953,726.19	$1,024,264.48	$1,067,823.51	26.505
CM SHO	$1,053,607.39	$1,070,217.58	$1,079,934.93	18.616
Levy-Flight SHO	$1,054,203.82	$1,068,160.76	$1,077,662.18	1.206

"-": means that is not available in the literature

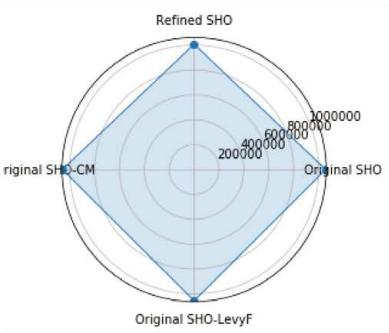
Table 8.- Load demand for 24 hours test case 2.

The figures in appendix section AF5 through AF8 shows a graphical representation of the detailed results of the power generated by each generating unit, for each algorithm compared.

Test case 3: 10-unit system neglecting power transmission loss.

For this test case, the valve points effects, the ramp-rate limits are considered, while neglecting the power transmission losses. The input data for the 10-unit system will be the same used in test case 2, enlisted in table 5, the load demand for 24 hours is the same for the test case 2, the figure 7 displays the average total dispatch cost of each SHO version used.

Figure 8 shows the fuel cost behavior over the iterations of the algorithms which used the same iterations, while figure 9 shows the fuel cost behavior of the LF-SHO.



g. 7.- Average total cost ($) of test case 3.

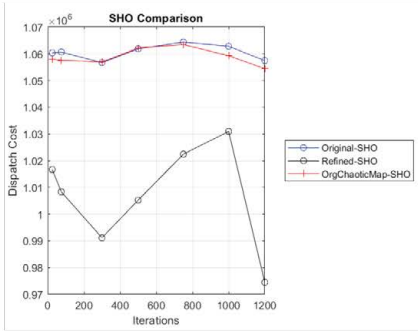


Fig. 8.- Average cost through iterations for test case 3.

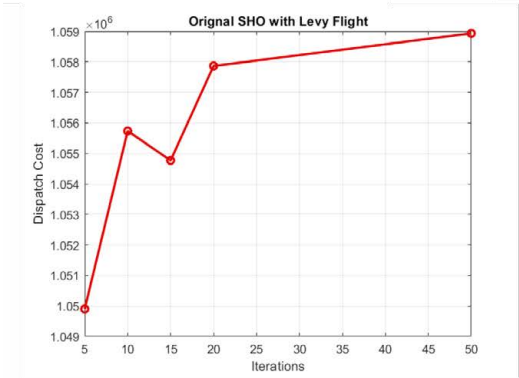


Fig. 9.- Average cost through iterations of the original SHO with Levy-Flight for test case 3.

From Table 9 it is observed that the minimum cost is $955,954.49 computed with the Refined SHO while the fastest algorithm is the LFSHO with an average time of 0.8 sec. this because the number of iterations. In the Appendix section are the results of each algorithm, the table A9 displays the result of the test case for the original SHO, the table A10 displays the results of the refined SHO while table A11 displays the results of the chaotic SHO and table A12 displays SHO with Levy-Flight operator, the figures AF9 through AF12 show a graphical representation of these results.

Method	Minimum Cost($)	Average Cost ($)	Maximum Cost($)	Time in secs
CMAES	$1,023,740.00	$1,026,307.00	$1,032,939.00	37.56
GSA	$1,019,455.18	$1,020,172.36	$1,020,635.15	61.24
CDBCO	$1,021,500.00	$1,024,300.00	-	40.2
ICPSO	$1,019,072.00	$1,020,027.00	-	28.02
Original SHO	$1,043,033.74	$1,058,676.04	$1,068,186.75	21.330
Refined SHO	$955,954.49	$1,007,706.10	$1,039,499.78	21.602
CM SHO	$1,046,677.84	$1,059,576.63	$1,071,506.78	23.359
LF SHO	$1,038,643.87	$1,055,167.21	$1,066,825.56	0.8013

Table 9.- Load demand for 24 hours test case 3.

6. Conclusions and future work

In this paper the Selfish Herd Optimizer as well as all the versions of the algorithm so far were implemented for the first time on a real engineering constrained optimization problem such as the dynamic economic dispatch.

All the versions proposed reach their goal and offers different improvements between them such as the minimum value, or the number of iterations needed to reach the local optima. In future, these algorithms could be implemented in some other versions of the economic dispatch problem like the multi-objective economic dispatch (MOED), or problems of another kind like image segmentation or combinatorial optimization.

7. References

[1].-W. Ongsakul, D.N.Vo,"Artificial Intelligence in Power System Optimization", Boca Raton: CRC Press,2013,p. 11.

[2].-B. Mandal, P.Kumar Roy, S. Mandal. (2014). Economic load dispatch using krill herd algorithm. *Electrical Power and Energy Systems.* 57,1-10.

[3].-D.Santra et. al. (2020) Dynamic economic dispatch using hybrid metaheuristics. *Journal of Electrical Systems and Information Technology.*7,1-30.

[4].-Jabr RA., Coonick AH., Cory BJ. (2000). A homogeneous linear programming algorithm for the security constrained economic dispatch problem. *IEEE Trans Power Syst*,15,930–6.

[5].-Chen C-L. (2007). Non-convex economic dispatch: a direct search approach. *Energy Convers Manage*.48,219–25.

[6].-Chandram K.,Subrahmanyam N., Sydulu M.(2008).Brent method for dynamic dispatch with transmission losses.*Paper presented at the IEEE/PES transmission and distribution conference and exposition. Chicago.*

[7].-Hindi K.,Ghani M.(1991).Dynamic economic dispatch for large-scale power systems:a Lagrangian relaxation approach.*Elect Power Syst Res*.13(1),51–6.

[8].-Coelho LDS., Mariani VC.(2008).Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects. *Energy Convers Manage.49(11),*3080–5.

[9].-Secui DC.(2015).A method based on the ant colony optimization algorithm for dynamic economic dispatch with valve-point effects. *Int Trans Electr Energy Syst 25(2),*262–287.https ://doi.org/10.1002/etep.1841.

[10].- Chiang C-L.(2005).Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE Trans Power Syst.20*,1690–9.

[11].- Yuan X., Wang L., Yuan Y., Zhang Y., Cao B., Yang B.(2008).A modified differential evolution approach for dynamic economic dispatch with valve-point effects.*Energy Convers Manage.49*,3447–53.

[12].- Roy PK., Mandal B., Bhattacharya K.(2012).Gravitational search algorithm based optimal reactive power dispatch for voltage stability enhancement. Electr Power Compon Syst.40(9),956–76.

[13].- Y. Wang et al.(2010). Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-points effects. *Energy Conversion and Management.51,*2893-2900.

[14].- P. Lu et al. (2014). Chaotic differential bee colony optimization algorithm for dynamic economic dispatch problem with valve-points effect. *Electrical Power and Energy Systems.62,*130-143.

[15].- Y. Lu et al. (2011). Chaotic differential evolution methods for dynamic economic dispatch with valve-points effects. *Engineering Applications of Artificial Intelligence.24,*378-387.

[16].- Han X, Gooi H. (2007). effective economic dispatch model and algorithm. *Int J Electr Power Energy Syst*; 29,113-20.

[17].- F. Fausto et al. (2017) A global optimization algorithm inspired in the behavior of selfish herds. *BioSystems*. 160. 39-55.

[18].- Hamilton, W.D., 1971. Geometry ot the selfish herd. J. Theor. Biol. 31 (2), 295–311.

[19].- Thomas, B., 1996. Evolutionary Algorithms in Theory and Practice. OxfordUniversity Press, Inc.Viscido, S.V., Wethey, D.S.

[20].- A. Yimit, K. ligura, Y. Hagihara. (2020). Refined selfish herd optimizer for global optimization problems. *Expert Systems With Applications*. 139.112838.

[21].- P. Anand, S.Arora.(2019).A novel chaotic selfish herd optimizer for global optimization and feature selection. Artificial Intelligence Review.https://doi.org/10.1007/s10462-019-09707-6

[22].- Gandomi A., Yang X-S., Talatahari S., Alavi A. (2013). Firefly algorithm with chaos. Commun Nonlinear *SciNumer Simul 18(1),*89–98.

[23].- Kohli M., Arora S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. JComput Des Eng 5(4),458–472.

[24].- Arora S., Singh S. (2017). An improved butterfly optimization algorithm with chaos. *J Intell Fuzzy Syst 32(1),*1079–1088.

[25].- He D., He C., Jiang L-G., Zhu H-W., Hu G-R.(2001). Chaotic characteristics of a one-dimensional iterative map with infinite collapses. *IEEE Trans Circuits Syst I Fundam Theory Appl 48(7)*, 900–906.

[26].- R. Zhao et al. (2020). Modified Selfish Herd Optimizer for Function Optimization. *International Journal of Computational Intelligence and Applications.* 19.2050003.

[27].- Q. X. Lieu, D. T. T. Do and J. Lee. (2018). An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Comput. Struct. 1(15),* 99–112.

[28].- M. Tomassini. (2016). Levy flights in neutral fitness landscapes, *Phys. A Stat. Mech. Appl. 4(15)*, 163–171.

[29].- N.E. Humphries, K.M. Schaefer, D. W. Fuller, G.E.M. Phillips, C. Wilding, D. W. Sims. (2016). Scale-dependent to scale-free: Daily behavioral switching and optimized searching in a marine predator. *Animal Behav. 3(13)*, 189–201.

[30].- M. Tomassini and A. Antonioni. (2015). Levy flights: Levy flights and cooperation among mobile individuals. *J. Theor. Biol. 1(7),* 154–161.

[31].- A. Reynolds.(2015). Liberating Levy walk research from the shackles of optimal foraging, *Phys. Life Rev. 9(14),* 59–83.

[32].- P. Anselme, T. Otto, O. guntnrkun.(2018). Foraging motivation favors the occurrence of Levy walks. *Behav. Process. 2(147)*, 48–60.

[33].- B. Yan, Z. Zhao, Y. Zhou, W. Yuan, J. Li, J. Wu, D. Cheng.(2017). A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters. *Comput. Phys. Commun. 10(219)*, 79–86.

[34].- P. S. Manoharan , P. S. Kannan , S. Baskar , M. Willjuice Iruthayarajan & V. Dhananjeyan.(2009).Covariance matrix adapted evolution strategy algorithm-based solution to dynamic economic dispatch problems, *Engineering Optimization*, *41:7*, 635-657, DOI:10.1080/03052150902738768

[35].- H. Maskani, M. Ashouri, B. et al. (2012). Gravitational Search Algorithm Optimization for Dynamic Economic Load Dispatch with Valve-Point Effects. *International Review on Modelling and Simulations.5,N.1*