



# A novel hybrid PSO–GWO algorithm for optimization problems

Fatih Ahmet Şenel<sup>1</sup> · Fatih Gökçe<sup>1</sup> · Asım Sinan Yüksel<sup>1</sup> · Tuncay Yiğit<sup>1</sup>

Received: 10 August 2018 / Accepted: 23 November 2018 / Published online: 3 December 2018  
© Springer-Verlag London Ltd., part of Springer Nature 2018

## Abstract

In this study, we propose a new hybrid algorithm fusing the exploitation ability of the particle swarm optimization (PSO) with the exploration ability of the grey wolf optimizer (GWO). Our approach combines two methods by replacing a particle of the PSO with small possibility by a particle partially improved with the GWO. We have evaluated our approach on five different benchmark functions and on three different real-world problems, namely parameter estimation for frequency-modulated sound waves, process flowsheeting problem, and leather nesting problem (LNP). The LNP is one of the hard industrial problems, where two-dimensional irregular patterns are placed on two-dimensional irregular-shaped leather material such that a minimum amount of the material is wasted. In our evaluations, we compared our approach with the conventional PSO and GWO algorithms, artificial bee colony and social spider algorithm, and as well as with three different hybrid approaches of the PSO and GWO algorithms. Our experimental results reveal that our hybrid approach successfully merges the two algorithms and performs better than all methods employed in the comparisons. The results also indicate that our approach converges to more optimal solutions with fewer iterations.

**Keywords** Exploitation · Exploration · Grey wolf optimizer (GWO) · Leather nesting problem (LNP) · Particle swarm optimization (PSO)

## 1 Introduction

Optimization problem is a common issue of many engineering domains, where an optimal solution is searched inside a complex space. When the problem is not solvable analytically or it is solvable but requires too much time, numerical approaches could help, although a globally optimized solution is not guaranteed. Meta-heuristic algorithms are among these numerical approaches taking their inspiration usually from the nature [1]. Besides developing a new meta-heuristic algorithm, another common approach is to

hybridize different algorithms to merge powerful sides of the algorithms with the aim of getting a better algorithm.

The particle swarm optimization (PSO) algorithm is one of the most successful optimization algorithms frequently used in the literature. As reviewed below, there are many studies utilizing the PSO algorithm to solve various optimization problems successfully due to its simplicity, small number of parameters and fast convergence speed. Since the PSO algorithm is very successful on various problems from different domains, we select it as one of the algorithms in this hybridization study. When we examine the PSO algorithm, its exploitation ability is very high but its exploration ability is poor [2–5]. By exploitation, we mean that the algorithm is very successful to perform a local search. Exploration ability makes an algorithm better at finding good starting positions, possibly near the global minimum. A good optimization algorithm is required to balance exploitation and exploration [2]. High exploration ability makes the PSO algorithm to suffer from converging to local minimum when its starting position is far away from global minimum [2–5]. To reduce the chance of trapping into a local minimum, the PSO algorithm do not directly accept new candidate positions. Instead, it prefers to chose random positions with a

✉ Fatih Ahmet Şenel  
fatihsenel@sdu.edu.tr

Fatih Gökçe  
fatihgokce@sdu.edu.tr

Asım Sinan Yüksel  
asimyuksel@sdu.edu.tr

Tuncay Yiğit  
tuncayyigit@sdu.edu.tr

<sup>1</sup> Department of Computer Engineering, Süleyman Demirel University, Isparta, Turkey

small possibility for some particles while the algorithm is proceeding on its iterations. However, this approach may be very risky since a random position may coincide with a terrible point in the search space. Our aim in this study is to improve this downside of the PSO algorithm. For this purpose, we need an algorithm with a good exploration ability. At this point, we utilized the recently proposed grey wolf optimizer (GWO) algorithm [6], since it is reported as starting with a high exploration [7]. In our approach, the GWO algorithm is used to support the PSO algorithm by replacing some particles with the new ones obtained by running the GWO algorithm for a small number of iterations. Hence, instead of jumping to a risky random position (which may result in trapping into a local minima or slowing down the optimization before reaching global minimum), we aim to find a better alternative by making use of exploration ability of the GWO algorithm.

In the literature, there are many hybrid approaches using meta-heuristic optimization methods. Below, we first review the literature proposing various hybrid approaches combining various methods and then the literature hybridizing only the PSO algorithm with different methods.

Mahmood et al. developed a hybrid GA method to optimize the real-time task allocation problem aiming to minimize the power consumption of DVS-enabled multiprocessor systems [8]. They have developed their hybrid method by combining the exploitation of the stochastic evolution algorithm with exploration ability of the GA. They proposed a specialized crossover operator and also defined a perturb operator to replace the mutation operator in the GA. Liu et al. presented a hybrid approach by combining ant colony optimization (ACO) algorithm and the GA algorithm [9]. They transferred the ants to the new generation using crossover, mutation and selection operators of the GA. They tested their method with optimal path selection problem. Torkaman et al. developed a hybrid approach using the GA and the SA methods [10]. The initialized population with the GA was improved using the SA to achieve a high fitness value. Lam et al. used the GA with the gradient method [11]. Since the GA was sufficient in the global search but insufficient in the local search, they integrated the gradient method which was successful in the local search. They started the search using the GA method to reduce to the local search area, and then quickly achieved the optimum position with the gradient method. In a similar approach, Yuce et al. developed a hybrid method to enhance the weakness of the bees algorithm (BA) in the global search combining it with the GA method [12]. The GA algorithm was forced to run the global search again at the best locations found by the BA. They tested their hybrid approach on the single machine scheduling problem. Vosoughi and Darabi developed a hybrid approach combining the conjugate gradient (CG) and the GA methods [13]. First, they divided the problem space into

evenly spaced layers. Then, they found the point of each layer that had the highest fitness value by the CG method. Finally, they performed a curved line/plane matching process through the best locations of all layers. Based on this curved line/plane, the entire search space was divided into several sub-spaces. Each sub-space was ranked in ascending order of fitness values and the GA method was activated. The GA method run the search process starting from the lowest sub-space and continued until the stop criteria was met. Chen and Huang developed a hybrid approach by combining the extension matrix (EM) and the GA for feature selection [14]. They first established an integer programming model using the EM, and then applied the GA to find an optimal feature subset. They showed the working of their method by applying it to intrusion detection problem. During the training phase of the detector, optimal features were selected with their hybrid approach.

Mafarja and Mirjalili developed two different hybrid optimization models by combining the SA method with the whale optimization algorithm (WOA) in two different ways [15]. In the first model, they used the SA method as an operator of the WOA method. In the WOA method, the best position was achieved by optimizing random movements with the SA. In the second model, the position with the highest fitness value was first found by the WOA, and then a final local search was performed with the SA method to find the best location. They have tested their hybrid approach with the feature selection process. Taheri et al. developed a hybrid artificial neural network (ANN)–artificial bee colony (ABC) algorithm that estimates ground vibration during mine blasting [16]. The ANN's weight and bias values for the network structure were determined by the ABC algorithm. Xue et al. developed a hybrid approach aiming to improve the ABC algorithm [17]. They used differential evolution (DE) algorithm in the generation of the new population where the ABC is weak. They improved the performance of the ABC algorithm, using the candidate solution generation strategy.

Singh and Singh used the GWO and sine cosine algorithm (SCA) together [18]. They optimized the position changes of the wolves with the SCA by removing random movements of alpha wolves in the GWO algorithm. Pan et al. developed a hybrid algorithm using the GWO and flower pollination algorithm (FPA) [19]. They updated the position of the wolves in the GWO algorithm using the position updates of the FPA algorithm.

Javidrad and Nazari developed a hybrid approach using the PSO algorithm and simulated annealing (SA) algorithms [20]. They applied a turn-based switching approach between the algorithms to combine them. The optimization started with the PSO algorithm, and was switched to the SA algorithm when the PSO could not find a better result than its previous iteration. While the SA algorithm was continuing the optimization, the turn passed to the PSO algorithm when

the SA algorithm was not able to find a better fitness value. This iterative switching process between two algorithms continued until the desired number of iterations was reached or a stop criteria was met.

Kumar and Vidyarthi developed another hybrid approach by mixing the GA and the PSO methods [21]. They aimed to find the optimum location by starting the search process with the PSO method to reduce the search domain to a local region, and then proceeding with the GA method.

Garg proposed a hybrid approach by combining the strengths of the PSO and the GA methods [22]. The GA aims to transfer good individuals to the next generation. In the PSO, however, all individuals continue to take part in the population throughout the iterations. Garg fused two methods such that at each iteration the GA method prepared the next generation to be used by the PSO by applying crossing, mutation and selection processes.

In another study, Ali and Tawhid combined the PSO and the GA methods in three phases [23]. First, the PSO algorithm was utilized to find a stable fitness value. Second, they divided the current population into sub-populations and then increased the diversity within each of the sub-populations using the GA's crossover operator. In this phase, the GA's mutation operator prevented falling to local minima. In the last phase, the optimization was again switched to the PSO algorithm, and the PSO was repeated until a maximum number of iterations was reached or the stop criteria was met.

Kaur and Mahajan developed a hybrid method using ant colony algorithm (ACA) and the PSO algorithm [24]. In this method, the population was first developed with the ACA and then the optimization was continued by the PSO algorithm.

Ghasemi et al. combined the adaptive neuro-fuzzy inference system (ANFIS) and the PSO algorithm [25]. They optimized the ANFIS structure with the PSO to minimize the difference between the outputs obtained with the ANFIS training and the actual results.

Hasanipanah et al. used support vector regression (SVR) and the PSO method to predict air-overpressure caused by mine blasting [26]. The hyper-parameters of each kernel was computed by the PSO method. They tested the SVR method with three different kernels.

Similar to our study, there are hybrid methods employing the PSO and GWO algorithms. In Chopra et al. and Kamboj, the PSO and GWO algorithms were used at each iteration one after the other [27, 28]. The population obtained with one algorithm was utilized by the other one in the next iteration. Here, the aim was to improve the population by exploiting the strengths of the algorithms at each iteration separately. The difference between these two studies is that in [27], all of the population were being transferred from one method to the other at every iteration, however, in [28], the population of one method was being updated with the best

individual of the other method obtained in its last iteration. Singh and Singh conducted another hybrid study using the PSO and GWO algorithms [29]. Instead of running the algorithms one by one at each iteration, they run them in parallel using a mixture of the algorithms' governing equations.

When the method of Chopra et al. is examined, no method can focus on solving the problem alone due to different characteristics of the algorithms. The method of Kamboj also faces a similar situation such that, though the whole population is not transferred, algorithms affect each other by directing the other algorithm towards the best solution they obtained on their turn. Updating the population of one method with the best individual of other method may harm the stability of both methods and hence, instabilities may arise. Although, the method of Singh and Singh run both algorithms in parallel, it is similar to the method of Kamboj such that the governing equations of the PSO method were updated to utilize the best three individuals of the GWO directly instead of using its own global best particle. Similarly, this approach also may result in instabilities. We have included these three hybrid methods in our experiments to compare their performances with our hybrid approach. For the sake of simplicity, in the rest of the paper, we will call these three studies, namely [27], [28], and [29] as HC, HK and HS, respectively.

In our study, unlike the PSO and GWO hybrid approaches reported above, the optimization process is under control of the PSO method. Some individuals of the PSO algorithm which is selected with a small possibility are replaced by the best individual of the GWO algorithm. The GWO algorithm is run with a small population for a small number of iterations so that partially improved individuals are created. Since the GWO algorithm starts with a high exploration ability, these partially improved individuals have an important potential to avoid falling into a local minimum. With our hybrid approach, the stability of the PSO algorithm remains intact and gets support from the GWO algorithm in terms of exploration.

The performance of an optimization algorithm should be tested on some benchmark functions as well as on a real-life problem to assess its effectiveness. For these purposes, we evaluated our hybrid approach on several benchmark functions taken from the literature [30], on the optimal nesting problem of two-dimensional patterns which is encountered as a hard optimization problem in many sectors today, and as well as on other two different real-world problems. These two problems are the parameter estimation for frequency-modulated (FM) sound waves [31] and process flowsheeting problem [32].

When we consider the optimal nesting problem of two-dimensional patterns, at the forefront are the sectors such as leather, textiles, paper, glass and furniture. In the majority of these sectors, the desired patterns are cut out from a

spool-wounded material whose width is fixed and length is variable. However, unlike others, there is no specific shape of the material in the leather sector. Leather materials have irregular shapes and may include low-quality and non-usable regions with again irregular shapes. Therefore, two-dimensional nesting problem is more difficult in the leather sector than in other sectors. In addition, since the natural leather is a non-recyclable raw material, the nesting and cutting operations have to be done much more carefully than other sectors.

The leather materials used in this study are natural leather and have irregular shapes. It is aimed to place the two-dimensional irregular-shaped patterns on two-dimensional irregular-shaped material in the most suitable way to minimize wastage. In the literature, this problem is also known as two-dimensional irregular nesting problem [33] or two-dimensional cutting stock problem [34, 35]. Cherri et al. developed two different models for this problem and evaluate them in a real-world scenario for metal cutting industry. Their two models were based on direct trigonometry and no-fit polygon covering, respectively. They reported their no-fit polygon covering model as outperforming their direct trigonometry model and one other model based on no-fit polygon (NFP). In [34] and [35], the LNP problem in automotive industry to produce car seats was studied. Optimum placement of the patterns without any overlap was solved with the help of the NFP technique. Before placement, the patterns were grouped according to their shape, and they were placed by giving precedence to big and regular patterns. In [36], Crispin et al. aimed to place shoe patterns onto real leather using genetic algorithm (GA) and also utilizing the NFP technique. In another study [37], Yuping et al. studied placement of irregular-shaped patterns onto leather by developing a fast algorithm based on simulated annealing (SA). During the placement, they utilized shifting operation to prevent overlaps between the patterns and overflow of the patterns outside of the leather hide.

The main highlights of this study are as follows: (1) we have developed a new hybrid PSO–GWO optimization method. (2) We validated our method using five different benchmark functions taken from the literature. (3) We evaluated our method on 3 different real-world problems, namely parameter estimation for frequency-modulated sound waves [31], process flowsheeting problem [32] and the leather nesting problem [34, 35]. (4) We compared the performance of our algorithm with the PSO, GWO, ABC [38], social spider algorithm (SSA) [39] algorithms as well as with three different and recently proposed hybrid PSO–GWO algorithms [27–29].

The rest of the paper is organized as follows. Section 2 explains the employed optimization methods and our hybrid approach. In Sect. 3, we define the optimization problems utilized to evaluate our approach. Section 4 presents the

experimental results along with the discussions. Finally, Sect. 5 concludes the paper.

## 2 Optimization methods

The hybrid algorithm proposed in this study utilizes the PSO and GWO meta-heuristic methods. The PSO is a well-known and widely used method [40–43]. The GWO, although being relatively new in the literature, is again a meta-heuristic method reported to give successful results like the PSO algorithm [6, 44, 45]. A hybrid approach has been introduced using these two algorithms that yields successful results. We describe our method in detail in the following sections.

### 2.1 Particle swarm optimization (PSO)

The PSO is a population-based meta-heuristic optimization method developed by Kennedy and Eberhart [40]. It is inspired by the social behavior of bird flocking or fish schooling when searching for food. Initially, the initial population is generated randomly within the search domain in the PSO method. The best location of each particle and the position information of the best particle in the swarm are constantly kept in memory. All particles in the swarm update their positions using the following equations in each iteration:

$$\bar{x}_{n+1}^i = \bar{x}_n^i + \bar{v}_{n+1}^i, \quad (1)$$

$$\bar{v}_{n+1}^i = \omega \bar{v}_n^i + c_1 r_1 (\bar{p}_n^i - \bar{x}_n^i) + c_2 r_2 (\bar{p}_n^g - \bar{x}_n^i). \quad (2)$$

Here,  $i$  refers to the particle in the swarm.  $n$  is the iteration step carried out, and  $r_1$  and  $r_2$  values represent random numbers in the range  $[0, 1]$ .  $\omega$  is the inertia weight parameter. The coefficients  $c_1$  and  $c_2$  represent the optimization parameters,  $\bar{x}$  is the position vector,  $\bar{v}$  is the velocity vector, and  $\bar{p}^i$  is the best position information that  $i$ th particle has achieved and finally,  $\bar{p}^g$  represents the best position information available in the swarm.

In the PSO algorithm, the new position and velocity of a particle are not accepted with a small possibility; instead, it is replaced by a random position within the search space. The aim of this operation is to escape from the local minimums.

The search continues until an optimum result is achieved or a maximum predefined number of iterations is reached.

### 2.2 Grey wolf optimizer (GWO)

The GWO algorithm is inspired by the leadership hierarchy of grey wolves [6, 44, 45]. Grey wolves are at the top of the

food chain. There are four types of grey wolves within the leadership hierarchy. These are alpha, beta, delta and omega wolves. In the GWO algorithm, alpha wolves represent the solution with the best result. Beta and delta wolves represent the second and third best solutions in the population. Omega wolves are the best solution candidates. The GWO algorithm assumes that hunting is performed by alpha, beta and delta wolves, while omega wolves follow these wolves.

Grey wolves’ hunting includes the following three main parts: (1) Tracking, chasing, and approaching the prey. (2) Pursuing, encircling, and harassing the prey till it stops moving. (3) Attacking the prey. Encircling the prey is modeled mathematically in the following equations:

$$D = |C \times X_p(t) - X(t)|, \tag{3}$$

$$X(t + 1) = X_p(t) - A \times D. \tag{4}$$

Here,  $t$  is the number of instantaneous iterations,  $X_p$  is the position of the prey,  $X$  is the location of grey wolves, and  $A$  and  $C$  are the coefficients for the vectors. The coefficients  $A$  and  $C$  are calculated as shown below:

$$A = a \times (2 \times r_1 - 1), \tag{5}$$

$$C = 2 \times r_2. \tag{6}$$

Here, the number of  $a$  is linearly decreasing from 2 to 0, as the number of iterations decreases.  $r_1$  and  $r_2$  represent uniformly selected random numbers between [0,1].

Grey wolves are led by alpha wolves to detect the location of the prey. Sometimes beta and delta wolves help the alpha wolf. The GWO algorithm assumes that the best solution is the alpha wolf, and the second best and the third solutions are beta and delta wolves, respectively. For this reason, the other wolves in the population move according to the position of these three wolves. This is formulated mathematically in the following equations:

$$\begin{aligned} D_\alpha &= |C_1 \times X_\alpha - X(t)|, \\ D_\beta &= |C_2 \times X_\beta - X(t)|, \\ D_\delta &= |C_3 \times X_\delta - X(t)|. \end{aligned} \tag{7}$$

The values  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$  represent the best three wolves in each iteration, respectively.

$$\begin{aligned} X_1 &= |X_\alpha - a_1 D_\alpha|, \\ X_2 &= |X_\beta - a_2 D_\beta|, \\ X_3 &= |X_\delta - a_3 D_\delta|, \end{aligned} \tag{8}$$

$$X_p(t + 1) = \frac{X_1 + X_2 + X_3}{3}. \tag{9}$$

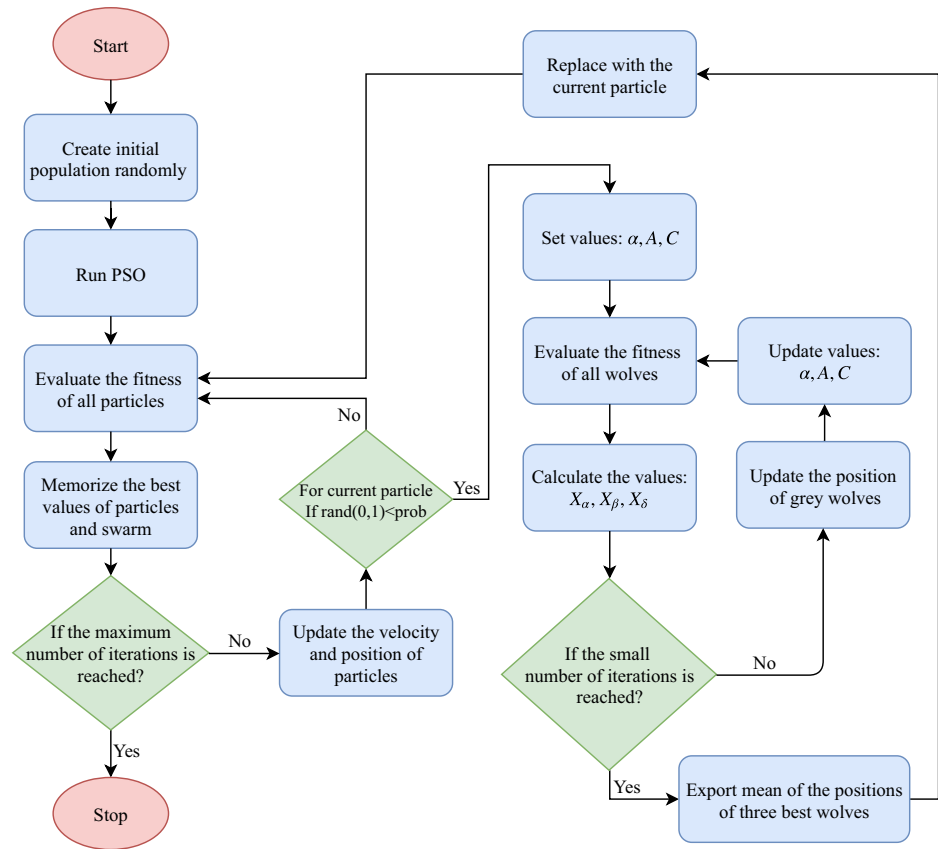
Here, the new position of the prey is expressed as  $X_p(t + 1)$  as the mean of the positions of three best wolves in the population.

Grey wolves finish the hunting by attacking the prey. For attacking, they must get close enough to the prey. When Eq. 5 is examined,  $A$  takes values that vary from  $[-2a, 2a]$  while  $a$  takes decreasing values from 2 to 0. When  $|A|$  value is greater than or equal to 1, existing hunts are abandoned to find better solutions. Assuming that the prey got close enough for values less than 1, grey wolves are forced to attack the prey. This approach prevents the wolves getting stuck on local minimum.

When the GWO algorithm reaches the desired number of iterations, the search is completed.

### 2.3 Hybrid PSO–GWO (HPSGWO)

Our Hybrid PSO–GWO algorithm (HPSGWO) has been developed without changing the general operation of the PSO and GWO algorithms. The PSO algorithm can achieve successful results in almost all real-world problems. However, a solution is needed to reduce the possibility of the PSO algorithm to trap into a local minimum. In our proposed method, the GWO algorithm is utilized to support the PSO algorithm to reduce the possibility of falling into a local minimum. As mentioned in Sect. 2.1, the PSO algorithm directs some particles to random positions with small possibility to avoid local minimums. As explained in Sect. 1, these directions may have some risks causing to move away from global minimum. The exploration ability of the GWO algorithm is used to prevent these risks by directing some particles to positions that are partially improved by the GWO algorithm instead of directing them to random positions. However, the running time is extended since the GWO algorithm is also employed in addition to the PSO algorithm. But, as will be presented in Sect. 4, when the success of the results and the amount of extra time needed are taken into consideration, the extended time can be regarded as tolerable depending on the optimization problem solved. The increased success can tolerate additional time requirement in sector such as leather where the wastage is much more critical as long as the solution is obtained in a feasible time (in minutes).

**Fig. 1** Flowchart of the HPSGWO**Algorithm 1** HPSGWO algorithm

---

```

1: MAXITR: the number of maximum iterations set by the user
2: PS: the number of population sizes set by the user
3: prob: small possibility rate set by the user
4: procedure HPSGWO
5:   Initialize particles
6:   for  $i = 1$  to MAXITR do
7:     for  $j = 1$  to PS do
8:       Run PSO
9:       Update the velocity and the position of current particle
10:      if  $\text{rand}(0, 1) < \text{prob}$  then ▷ to avoid from the local minima
11:        Set a, A, C values
12:        for  $k = 1$  to 10 do ▷ small number of iterations
13:          for  $m = 1$  to 10 do ▷ small number of population sizes
14:            Run GWO
15:            Update the position of  $\alpha, \beta, \delta$  wolves
16:            Update a, A, C values
17:          end for
18:        end for
19:        position of current particle = mean of the positions of three best wolves
20:      end if
21:    end for
22:  end for
23: end procedure

```

---

**Table 1** Definitions of the benchmark functions

Function no	Function description	Function name	Dimension	$F_{\min}$
$F_1$	$\sum_{i=1}^D \left( \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (z_i + 0.5))] \right)$ $- D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (z_i \times 0.5))] + f_{\text{bias}}$ $a = 0.5, b = 3, k_{\max} = 20, z = (x - o) \times M$ $x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D] x \in [-0.5, 0.5], f_{\text{bias}} = 90$	Shifted Rotated Weierstrass Function [30]	30	90.0
$F_2$	$\sum_{i=1}^D (z_i^2) + f_{\text{bias}}$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$ $x \in [-100, 100], f_{\text{bias}} = -450$	Shifted Sphere Function [30]	30	- 450.0
$F_3$	$\sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$ $x \in [-5, 5], f_{\text{bias}} = -330$	Shifted Rastrigin's Function [30]	30	- 330.0
$F_4$	$\sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}$ $z = (x - o) \times M, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$ $x \in [-5, 5], f_{\text{bias}} = -330$	Shifted Rotated Rastrigin's Function [30]	30	- 330.0
$F_5$	$\sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_{\text{bias}}$ $z = x - o, x = [x_1, x_2, \dots, x_D], o = [o_1, o_2, \dots, o_D]$ $x \in [-100, 100], f_{\text{bias}} = -450$	Shifted Schwefel's Problem 1.2 [30]	30	- 450.0

The algorithm and the flowchart of our HPSGWO method are given in Algorithm 1 and Fig. 1, respectively.

### 3 Definition of optimization problems

We evaluate our hybrid approach on five different benchmark functions ( $F_1, F_2, F_3, F_4$  and  $F_5$ ) and on three different real-world problems. Table 1 presents the definition of these benchmark functions which are taken from Congress on Evolutionary Computation (CEC) 2005 benchmark functions [30]. All of these functions have a dimension of 30. The global minimum values,  $F_{\min}$ , of these functions are also given in Table 1. The real-world problems utilized are parameter estimation for frequency-modulated sound waves (PEFMSW) [31], process flowsheeting problem (PFP) [32] and the leather nesting problem [34, 35]. The definitions for the PEFMSW and PFP are given in Table 2 together with their dimensions and  $F_{\min}$  values. We will also call the PEFMSW and PFP as  $F_6$  and  $F_7$ , respectively, in the rest of the paper. The following section defines the leather nesting problem.

### 3.1 Definition of the leather nesting problem

The leather nesting problem (LNP) is an optimization problem where desired patterns are required to be placed on a leather material with an objective of minimizing the wasted material. When the material is an imitation leather, the wastage can be easily recycled as in wood, glass, textile, and paper materials. However, when it is the natural leather, the wastage should be reduced to minimum, since the natural leather is a non-recyclable raw material. Unnecessary wastage brings big losses to large-scale companies. In addition to the non-recyclable characteristics of the natural leather, there are also some other issues which make the problem even harder: (1) natural leather does not have a regular shape, and can have tears and holes, and (2) the shoe patterns used in the production also do not have regular shapes.

The LNP problem is also named in the literature as a two-dimensional irregular nesting process [33–35].

The LNP problem aims to place shoe patterns ( $P = \{P_1, P_2, P_3, \dots, P_N\}$ ) with irregular shapes onto the irregular shaped leather material ( $M$ ) with minimal waste. Rotation of the patterns is allowed. However, overlap which

**Table 2** Definitions for two real-world problems, namely the PEFMSW and PFP

Function no	Function description	Function name	Dimension	$F_{\min}$
$F_6$	$\sum_{t=0}^{100} (y(t) - y_0(t))^2$ $y(t) = a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta)))$ $y_0(t) = (1.0) \sin((5.0)t\theta - (1.5) \sin((4.8)t\theta + (2.0) \sin((4.9)t\theta)))$ $\theta = 2\pi/100, x_i, \omega_i \in [-6.4, 6.35] \text{ for } i \in \{1, 2, 3\}$	Parameter estimation for frequency-modulated sound waves [31]	6	0.0
$F_7$	$-0.7x_3 + 5(x_1 - 0.5)^2 + 0.8$ $\begin{cases} g_1 = -\exp(x_1 - 0.2) - x_2 \leq 0 \\ g_2 = x_1 - 1.2x_3 - 0.2 \leq 0 \\ g_3 = x_2 + 1.1x_3 + 1.0 \leq 0 \end{cases}$ $x_1 \in [0.2, 1], x_2 \in [-2.22554, -1], x_3 \in \{0, 1\}$	Process flowsheeting problem [32]	3	1.0765

indicates that there are at least two patterns intersecting, and overflow which means that there is at least one pattern whose boundaries are exceeding the leather are not allowed. The following equation gives the definition of these two constraints:

$$\begin{aligned}
 P_i \cap P_j &= \emptyset & 1 \leq i, j \leq N \\
 P_i &\subseteq M & 1 \leq i \leq N.
 \end{aligned}
 \tag{10}$$

Here,  $N$  denotes the number of patterns that are placed. The problem to be minimized after defining the constraints is expressed as shown below:

$$\begin{aligned}
 \text{Minimize} & \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N \text{Overlap}(P_i, P_j) + \sum_{i=1}^N \text{Overflow}(P_i) + \text{Cost}(P_i) \\
 \text{subject to} & \quad P_i \cap P_j = \emptyset & 1 \leq i, j \leq N \\
 & \quad P_i \subseteq M & 1 \leq i \leq N.
 \end{aligned}
 \tag{11}$$

The acceptable solutions are those where the sum of the overlap and overflow values is zero. The cost function, given in the following equation, determines the quality of the solution in acceptable situations:

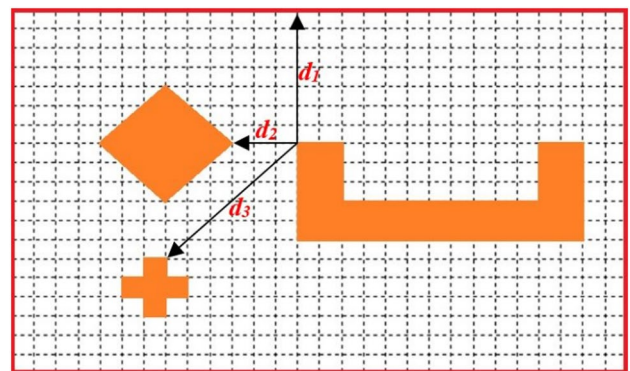
$$\text{Cost} = \sum_{i=1}^C \text{cost\_vertex}(i).
 \tag{12}$$

Here,  $C$  is the number of vertexes for a polygon (representing a pattern). While the cost\_vertex function calculates the cost of a vertex, first, the minimum distances from this vertex to the material boundary and all other polygons are calculated. Then, the shortest value in all calculated distances gives the cost of that vertex. Since the polygons are placed one by one, the cost function forces the placement polygons closer to other previously placed polygons and the bounds of the

material so that the wasted material is minimized. In Fig. 2, the calculation of cost for one of the vertexes of a polygon is illustrated. See Appendix for a detailed explanation of how the minimum distances between polygons are calculated.

### 4 Experimental results and discussion

To evaluate the performance of our hybrid optimization method and to compare it with the PSO, GWO, ABC, SSA, HC, HK and HS methods, we conducted experiments with



**Fig. 2** Calculation of Cost for one of the vertexes of U-shaped polygon. The minimum distances of the vertex point to other polygons and to the bounds of the material are as follows:  $d_1 = 7, d_2 = 3, d_3 = 6\sqrt{2}$ . The cost of the vertex point equals to  $\min(d_1, d_2, d_3) = d_2$  which is 3 units



five different benchmark functions and three different real-world problems presented in Sect. 3.

In all of the experiments, parameters of the PSO were selected as follows [46–49]:  $\omega = 0.7298$ , and  $c_1 = c_2 = 1.49445$ . Similarly, during the whole experiments, parameters of the SSA were set to: [50–52]:  $r_a = 1.0$ ,  $p_c = 0.7$  and  $p_m = 0.1$ .

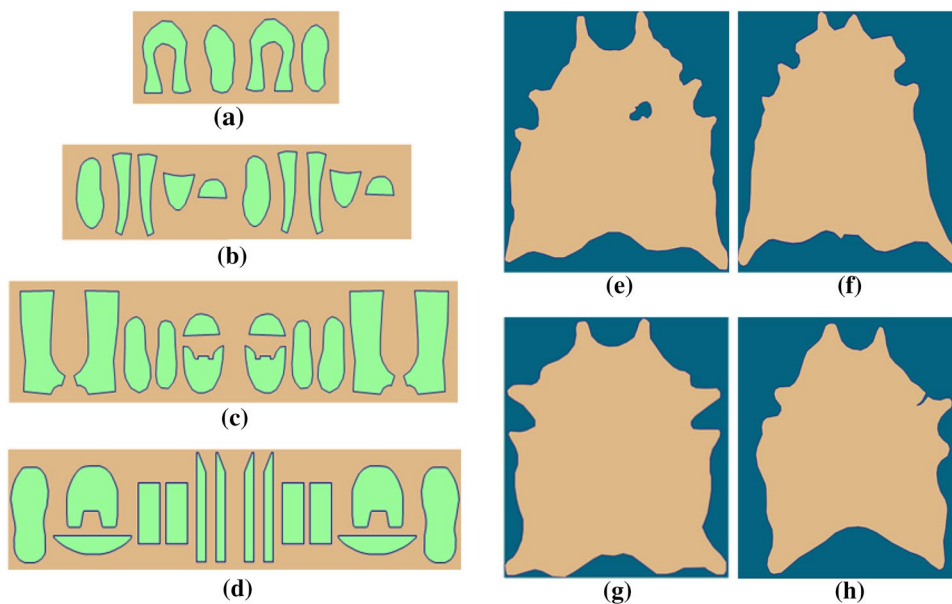
Experiments on  $F_1, F_2, F_3, F_4$  and  $F_5$  were repeated 25 times for each method and for each function. Experiments with  $F_6$  and  $F_7$  (PEFMSW and PFP) were conducted by applying each method 100 times to each problem. In these experiments, the methods are compared in terms of their best and worst results along with the mean and standard deviation statistics of their results. In the experiments performed with  $F_1 \sim F_7$ , we terminated the algorithms in each run when they reach a maximum number of  $10^4 \times D$  objective function evaluations. Here,  $D$  denotes the dimension of the corresponding function.

In the LNP experiments, we used four different shoe models with a total of 40 patterns and four different leather materials by representing them as polygons. Figure 3 shows these patterns belonging to real shoe models. The leather

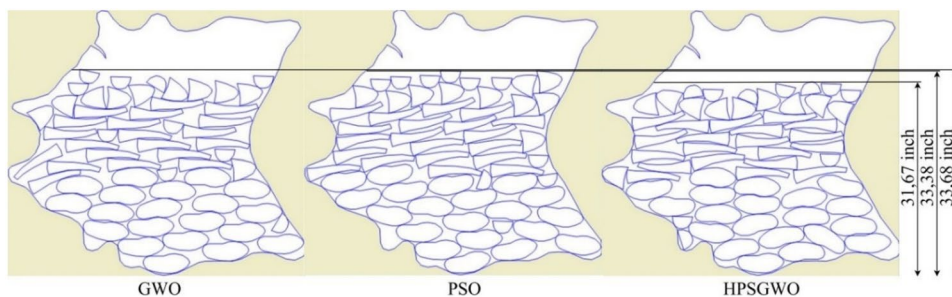
materials extracted from real hides and used in the evaluations are illustrated in again Fig. 3. We should note that all the patterns and materials have irregular shapes. During the evaluations, each of four shoe model patterns were placed onto four different materials separately. For each model and material pair, each optimization method was repeated five times. Furthermore, each optimization method was given with the same number of shoe patterns to be placed onto the current material. In these experiments, the population size is set to 50 and each method was run for 150 iterations in each test. However, since HK and HS methods resulted in very poor performance with default values, we had to run them by increasing the population size and the number of iterations to 150 and 450, respectively, only then we were able to obtain a comparable performance.

To evaluate the results of the LNP experiments, two different performance metrics are utilized, namely: (1) mean efficiency (ME) calculated as the average percentage of the occupied area of the placed patterns at the end of the optimization over the used area, and (2) mean placement time (MPT) calculated by averaging the required times to place all patterns in each of five different runs. During the

**Fig. 3** The patterns of the four different shoe models and leather materials used: **a** Model 1, **b** Model 2, **c** Model 3, and **d** Model 4 patterns; **e** Material 1, **f** Material 2, **g** Material 3, and **h** Material 4



**Fig. 4** The results of nesting for the GWO, PSO, and our HPSGWO methods using Model 2 and Material 4 are illustrated to explain the definition of used area necessary to calculate ME metric



**Table 3** Results of the experiments conducted with  $F_1, F_2, F_3, F_4$  and  $F_5$

Function name	Metrics	PSO	GWO	ABC	SSA	HC	HK	HS	HPSGWO
$F_1$	Best	105.7006	114.9096	121.7113	111.0104	114.2313	143.0604	128.9681	<b>102.9137</b>
	Worst	121.9920	126.4352	128.5150	119.7598	123.4563	156.2101	134.4481	120.8254
	Mean	112.8334	121.6975	125.7962	117.0926	119.0227	149.2469	131.2379	<b>112.7126</b>
	Std	3.8747	3.0525	1.5707	1.7962	2.2277	3.6614	1.4385	4.4379
$F_2$	Best	125.9714	1783.9904	2031.7467	- 427.3725	50.0422	21545.0398	56327.9534	- <b>450.0000</b>
	Worst	1841.3794	105138.4783	18486.3973	- 322.9848	575.4239	56821.2616	87825.3125	937.5272
	Mean	888.9429	6693.6779	9535.5799	- <b>385.7636</b>	365.7421	36361.7848	73613.2514	283.6992
	Std	501.9531	20099.7929	3934.2590	26.7099	131.4265	9479.0740	7553.2712	390.3624
$F_3$	Best	- 305.2286	- 182.8422	- 245.0817	- 93.3633	- 130.0256	1.1063	96.3287	- <b>317.0655</b>
	Worst	- 203.6240	- 120.3764	- 140.4272	- 56.6294	- 81.9586	345.6842	166.8426	- 230.9049
	Mean	- 264.0750	- 159.3470	- 179.3599	- 74.3561	- 108.9474	143.0156	129.6237	- <b>283.1083</b>
	Std	23.6911	14.4012	21.8508	10.3420	13.4839	65.9474	23.8654	18.8635
$F_4$	Best	- 233.6276	- 132.4912	126.9666	- 45.9506	- 136.6507	180.7946	248.7985	- <b>236.4742</b>
	Worst	- 114.8198	- 64.7051	326.2495	- 16.1413	- 36.4332	547.4909	491.7458	- 94.3715
	Mean	- <b>185.3406</b>	- 98.8701	242.8365	- 31.9284	- 87.8573	315.1692	396.0627	- 170.6654
	Std	30.9942	15.9753	56.9670	7.9571	25.8372	85.7930	52.4145	39.6899
$F_5$	Best	- 449.9946	5791.9056	45326.6877	2684.4428	706.9211	109313.7942	68222.8633	- <b>449.9995</b>
	Worst	3215.1047	22324.2271	75212.9893	5495.1434	2038.6187	2855039.8527	128822.8073	584.8864
	Mean	338.0734	9925.1390	61538.5074	4020.6433	1439.1417	939871.9197	90447.5096	- <b>143.0111</b>
	Std	816.6653	3735.4339	7488.1730	649.0649	364.6407	803748.8532	14636.8432	269.1572

Bold values indicate the best solutions found by each algorithm

**Table 4** Results of the experiments conducted with  $F_6$  and  $F_7$

Function name	Metrics	PSO	GWO	ABC	SSA	HC	HK	HS	HPSGWO
$F_6$	Best	<b>0.0000</b>	10.3444	6.6878	0.4444	7.8791	27.4715	24.2852	<b>0.0000</b>
	Worst	25.9758	25.1607	23.7307	20.1060	26.3509	130.877	30.4727	25.0971
	Mean	14.0708	18.5759	17.4936	14.0765	18.5890	41.6903	28.3669	<b>13.6978</b>
	Std	6.6699	4.3529	3.3131	3.3880	4.4055	17.7599	1.3863	5.4824
$F_7$	Best	<b>1.0765</b>	1.0941	1.0780	1.0767	1.0792	1.3942	1.0845	<b>1.0765</b>
	Worst	1.2500	1.2500	1.3305	1.3500	1.2500	3.2384	1.3484	1.2500
	Mean	1.1879	1.2357	1.1375	<b>1.1308</b>	1.1986	2.3627	1.2596	1.1771
	Std	0.0828	0.0410	0.0487	0.0554	0.0693	0.5877	0.0633	0.0856

Bold values indicate the best solutions found by each algorithm

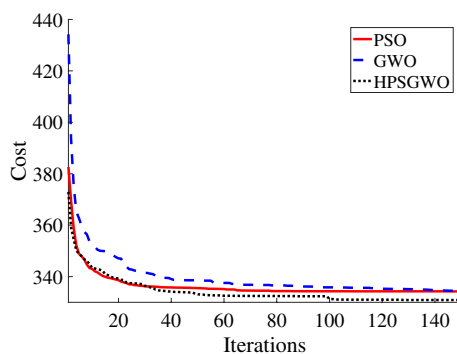
calculation of the ME, the used area is taken as the area extracted from the whole area of the material by removing its unused portion beginning from the boundary point of the farthest pattern along the placement direction. For instance, the used areas for three example placements shown in Fig. 4 are the areas below straight lines for each of them. Experiments related with MPT were all conducted on the same computer with Intel Core i5-6500 3.2 GHz processor and 4 GB of RAM.

In Table 3, the results of the experiments performed with  $F_1, F_2, F_3, F_4$  and  $F_5$  are illustrated. When the results are examined, our approach, HPSGWO, remarks as the only method achieving the best positions not reachable by other methods for all functions. The HPSGWO is able to find the global minimum for  $F_2$ , and its best position obtained for  $F_5$  is very close to the global minimum. In all aspects, our approach outperforms the GWO, ABC, HC, HK and HS methods.

The PSO is giving better results than the GWO according to Table 3. Our hybrid approach is performing better than the PSO in all functions except for  $F_4$ . This means that

**Table 5** Results obtained in the LNP experiments. Bold values indicate experiments that the HPSGWO algorithm is better than all other methods used. For ME, the higher is better; for MPT, the lower is better

Optimization method	PSO		GWO		ABC		SSA		HC		HK		HS		HPSGWO		
	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	ME (%)	MPT (s)	
Material 1	Model 1	50.46	70.11	50.66	71.64	46.79	114.00	49.01	125.00	49.65	106.22	43.15	1139.74	37.48	1253.65	<b>55.12</b>	193.98
	Model 2	64.29	238.64	62.23	238.17	51.46	353.00	57.79	342.50	57.94	271.41	51.06	2122.73	45.79	3055.06	<b>65.97</b>	395.50
	Model 3	61.85	100.00	63.69	99.75	55.04	158.64	59.60	150.96	60.45	115.63	53.90	1059.92	48.32	1459.21	<b>66.27</b>	182.50
	Model 4	63.79	161.82	63.28	155.98	47.91	285.18	53.39	301.98	58.39	255.31	55.27	2639.64	46.57	3713.10	<b>67.85</b>	382.70
Material 2	Model 1	55.41	61.92	53.70	61.02	46.10	106.60	49.45	117.80	51.03	96.24	48.56	1143.00	39.67	1152.17	<b>56.29</b>	142.18
	Model 2	66.03	227.33	66.54	236.69	52.94	349.5	54.62	305.00	60.32	246.57	55.22	2207.10	47.18	3197.50	<b>67.66</b>	371.67
	Model 3	66.11	93.46	66.51	91.20	57.22	134.4	63.24	133.44	62.65	104.72	56.69	1061.20	51.78	1378.41	<b>68.98</b>	158.38
	Model 4	64.16	151.23	63.20	150.31	53.51	271.32	56.87	264.60	59.71	246.6	58.93	3061.21	48.56	3330.16	<b>69.64</b>	358.40
Material 3	Model 1	54.86	63.06	52.99	64.57	43.23	105.00	51.22	102.80	50.29	99.57	43.50	1219.28	41.34	1321.64	<b>55.45</b>	150.24
	Model 2	67.32	231.53	65.80	241.10	50.51	360.00	59.19	360.00	59.92	259.06	54.56	2265.63	46.78	2997.89	<b>69.00</b>	386.32
	Model 3	65.78	99.42	67.14	96.74	51.96	166.32	59.31	151.44	63.36	110.30	63.05	1074.73	51.22	1475.13	<b>69.78</b>	168.52
	Model 4	65.54	168.07	66.6	156.21	51.81	300.30	60.68	310.38	61.18	249.20	61.21	2905.53	47.23	3422.55	<b>69.52</b>	374.23
Material 4	Model 1	54.48	62.71	54.36	59.52	45.86	110.60	48.76	105.60	51.06	95.13	47.84	1101.73	42.81	1307.99	<b>55.74</b>	142.39
	Model 2	65.53	223.74	65.26	222.26	52.71	291.50	57.46	350.50	59.57	249.51	50.27	2067.82	46.71	3129.84	<b>68.24</b>	371.66
	Model 3	64.08	98.75	65.86	96.20	53.16	127.20	58.50	118.32	63.45	105.36	56.81	1047.87	52.36	1461.80	<b>67.10</b>	159.91
	Model 4	64.75	165.05	66.49	149.47	50.40	301.14	56.28	265.02	60.71	242.86	57.56	3313.38	47.29	3521.52	<b>69.34</b>	358.81



**Fig. 5** The convergence comparison graph for the placement of a single sample pattern

via hybridizing the PSO with a less performing method, we are getting a better method than both of the PSO and GWO which clearly shows the success of our hybridization strategy. From this successful result, we can deduce that we managed to improve reportedly poor exploration ability of the PSO by merging it with the GWO.

There are only two cases where HPSGWO falls behind only the PSO and SSA methods in terms of only mean value. The SSA and PSO give better mean results than the HPSGWO only for  $F_2$  and  $F_4$ , respectively. However, when we consider best values, our method is able to reach better positions than these two methods indicating the improved exploration ability of our approach.

Table 4 presents the results of the experiments conducted with  $F_6$  and  $F_7$ . The HPSGWO and PSO are the only two methods able to reach the best values which are also global minimums. In this case, we can compare two methods by looking at the mean values. In terms of mean values, the HPSGWO is giving better results than the PSO in both problems. This indicates that via the exploration contribution of the GWO, our approach is able to obtain better results on the average. Although the SSA results in better mean value for  $F_7$  than the HPSGWO, it is not able to find global minimum which is achieved by our approach.

The results of the LNP experiments are given in Table 5. When the mean efficiency is considered, the HPSGWO algorithm is more successful than all of the other methods. When we calculate the average ME increases<sup>1</sup> over all experiments, the HPSGWO performs better than the PSO, GWO, ABC, SSA, HC, HK and HS with 4.77%, 4.85%, 28.55%, 16.39%, 12.05%, 21.87% and 40.75%, respectively. When we consider the SSA, although it gives comparable results

<sup>1</sup> These values are calculated by first taking the ME differences between the HPSGWO and other algorithms, then converting these differences to the percentages with respect to minimum ME value, and finally taking the average of these percentages.

for  $F_1 \sim F_7$ , for the LNP its ME results are always behind our approach. The low performances of the HK and HS even with an increased population size and number of iterations indicate that they are not suitable for the LNP.

If we examine the visual placement results in Fig. 4 given to better illustrate the calculation of our ME metric, the better performance of the HPSGWO is seen when the difference between the straight lines is inspected. The HPSGWO managed to place all of the patterns into a smaller used area, hence leaving a larger remaining area for a later use. The PSO and GWO methods have produced similar results to each other. We can deduce from these placements that the PSO and GWO trap into local minimums while our approach achieves a better result.

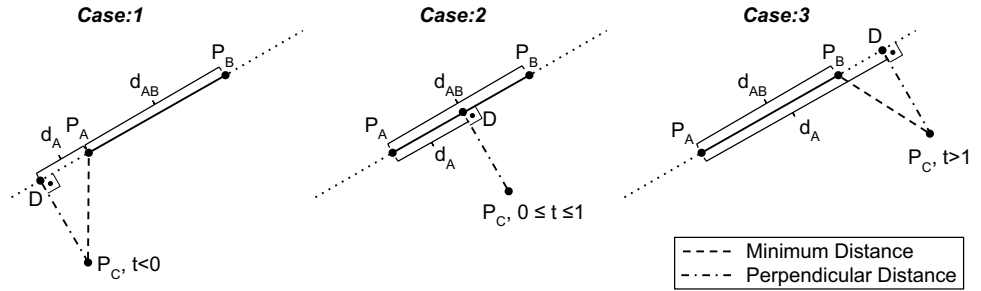
When the mean placement times in Table 5 are considered, the results show that the HPSGWO performs better than the other hybrid methods but not better than the PSO, GWO, ABC and SSA. When we consider that our approach utilizes the GWO whenever needed without modifying the governing equations of the PSO and GWO, this increased time complexity is inevitable. In the worst case, 2.77 times (for Material 1 and Model 1), in the best case, 1.63 times (for Material 2 and Model 2), and on the average 2.07 times more time is needed. However, when we consider the enhancement in the ME, the longer time need is not intolerable as the order of the placement time is in minutes. Since the natural leather is not recyclable, the decreased leather wastage is more critical for the leather industry.

Additionally, in Fig. 5, we present average convergence of only the PSO, GWO and HPSGWO methods to better illustrate the success of our hybrid approach. This convergence graph is depicted by running three methods on Model 4 and Material 4. For each method and for each iteration in Fig. 5, the cost of all patterns is averaged. We should note that, in the experiments, each pattern is placed onto the material one by one, and methods are iterated 150 times to place one pattern. This result clearly indicates that our hybrid approach converges to a lower cost value with less iterations than the PSO and GWO. Hence, we can conclude that our approach successfully enhances the exploration ability of the PSO.

## 5 Conclusion

In this paper, a new hybrid algorithm has been presented using the PSO and GWO algorithms. Our approach aims to prevent the PSO algorithm from falling into local minimums by utilizing the exploration ability of the GWO algorithm. For this purpose, some particles in the PSO algorithm were replaced with a small possibility by a partially improved particle obtained via the GWO algorithm. We have tested the performance of our algorithm and compared it with the PSO, GWO, ABC, SSA meta-heuristic methods, and as well as with

**Fig. 6** Three different cases for a point,  $P_C$ , and an edge,  $P_A B$



three different hybrid approaches of the PSO and GWO proposed in the literature. To evaluate our method comprehensively, we conducted experiments with five different benchmark functions and three different real-world problems. The results of the experiments indicate that our hybrid approach performs better than the conventional PSO, GWO algorithms, ABC, SSA and three hybrid PSO–GWO approaches. Since our hybrid approach do not modify the governing equations of the methods and apply the GWO with a small probability in the main flow of the PSO, its time complexity is higher than the PSO and GWO. However, our average convergence experiment demonstrate that our approach converges to lower cost values with few number of iterations. Additional time complexity of our approach should be considered depending on the problem it is applied, since in problems such as the LNP, obtaining more optimal solutions is much more critical as long as the increased time complexity is feasible.

### Appendix

In Sect. 3.1, we have given Eq. 12 for the calculation of cost and Fig. 2 for illustrating the procedure with sample polygons. In this appendix, we give the details of distance calculations needed to compute the cost.

A polygon consists of multiple edges. To calculate minimum distance of a vertex point to a polygon,  $d_{min}$ , we need to calculate the minimum distances from that point to all different edges of the polygon. The minimum of these distances gives us  $d_{min}$ . For this reason, we will below explain how the minimum distance of a vertex point to one of the edges is calculated.

In Fig. 6, we depicted three different cases for a point,  $P_C$ , and an edge,  $P_{AB}$ , where the intersection of the perpendicular line from  $P_C$  to  $P_{AB}$  falls into three different sections, namely: (1) outside the  $P_{AB}$  and close to  $P_A$ , (2) over  $P_{AB}$ , and (3) outside the  $P_{AB}$  and close to  $P_B$ . Here, we define a parameter  $t$  whose magnitude is calculated by dividing  $d_A$  to  $d_{AB}$ , but we determine its sign as follows. When the intersection of the perpendicular line is outside the  $P_{AB}$  and close to  $P_A$ ,  $t$  is negative, for the other cases  $t$  is positive. In other words,  $P_A$  is considered as a reference point. We can formulate the

coordinates of the perpendicular intersection point,  $D$ , in terms of  $t$  and coordinates of points  $P_A$  and  $P_B$  as in Eq. 13. Our aim is to find  $t$ , and for this purpose, we first express the distance,  $|DP_C|$ , between  $D$  and  $P_C$  as shown below:

$$\begin{aligned} D_x &= P_{Ax} + t \times (P_{Bx} - P_{Ax}), \\ D_y &= P_{Ay} + t \times (P_{By} - P_{Ay}), \end{aligned} \tag{13}$$

$$|DP_C| = \sqrt{\left( (P_{Ax} + t \times (P_{Bx} - P_{Ax})) - P_{Cx} \right)^2 + \left( (P_{Ay} + t \times (P_{By} - P_{Ay})) - P_{Cy} \right)^2} \tag{14}$$

Since  $DP_C$  segment is perpendicular to the given edge only when  $|DP_C|$  is minimum, we can find an equation to calculate  $t$  by minimizing  $|DP_C|$ . To minimize  $|DP_C|$ , we can take the first derivative of Eq. 14 in terms of  $t$  and equate it to zero. For the sake of simplicity, we can use the derivative of the terms inside the square root sign, since the minimization of those terms will also minimize  $|DP_C|$ . Such a derivation will result in Eq. 15. When we equate Eq. 15 to zero, we find Eq. 16 for  $t$ .

$$\begin{aligned} \frac{d}{dt} |DP_C|^2 &= 2 \times ((P_{Ax} + t \times (P_{Bx} - P_{Ax})) - P_{Cx}) \times (P_{Bx} - P_{Ax}) \\ &\quad + 2 \times ((P_{Ay} + t \times (P_{By} - P_{Ay})) - P_{Cy}) \times (P_{By} - P_{Ay}), \end{aligned} \tag{15}$$

$$t = \frac{(P_{Cx} - P_{Ax}) \times (P_{Bx} - P_{Ax}) + (P_{Cy} - P_{Ay}) \times (P_{By} - P_{Ay})}{(P_{Bx} - P_{Ax})^2 + (P_{By} - P_{Ay})^2} \tag{16}$$

Given a point  $P_C$  and an edge, after calculating  $t$  with Eq. 16, we can find minimum distance of  $P_C$  to the edge,  $d_{C_{min}}$ , using Eq. 17. Note that, for the Cases 1 and 3 in Fig. 6, closest distances are to the points  $P_A$  and  $P_B$ , respectively.

$$d_{C_{min}} = \begin{cases} \sqrt{(P_{Ax} - P_{Cx})^2 + (P_{Ay} - P_{Cy})^2} & t < 0 \\ \sqrt{\left( (P_{Ax} + t \times (P_{Bx} - P_{Ax})) - P_{Cx} \right)^2 + \left( (P_{Ay} + t \times (P_{By} - P_{Ay})) - P_{Cy} \right)^2} & 0 \leq t \leq 1 \\ \sqrt{(P_{Bx} - P_{Cx})^2 + (P_{By} - P_{Cy})^2} & t > 1. \end{cases} \tag{17}$$

## References

- Christensen J, Bastien C (2016) Chapter—seven-heuristic and meta-heuristic optimization algorithms. In: Christensen J, Bastien C (eds) *Nonlinear optimization of vehicle safety structures*. Butterworth-Heinemann, Oxford, pp 277–314
- Yang X-S (2014) *Swarm intelligence based algorithms: a critical analysis*. *Evolut Intell* 7:17–28
- Arani BO, Mirzabeygi P, Panahi MS (2013) An improved pso algorithm with a territorial diversity-preserving scheme and enhanced exploration-exploitation balance. *Swarm Evolut Comput* 11:1–15
- Wang Y, Li B, Weise T, Wang J, Yuan B, Tian Q (2011) Self-adaptive learning based particle swarm optimization. *Inf Sci* 181(20):4515–4538 (special issue on interpretable fuzzy systems)
- Kessentini S, Barchiesi D (2010) A new strategy to improve particle swarm optimization exploration ability. 2010 Sec WRI Glob Congress Intell Syst 1:27–30
- Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. *Neural Comput Appl* 30:413–435
- Mahmood A, Khan S, Albalooshi F, Awwad N (2017) Energy-aware real-time task scheduling in multiprocessor systems using a hybrid genetic algorithm. *Electronics* 6:40
- Liu J, Xu S, Zhang F, Wang L (2017) A hybrid genetic-ant colony optimization algorithm for the optimal path selection. *Intell Autom Soft Comput* 23:235–242
- Torkaman S, Fatemi Ghomi S, Karimi B (2017) Hybrid simulated annealing and genetic approach for solving a multi-stage production planning with sequence-dependent setups in a closed-loop supply chain. *Appl Soft Comput* 71:1085–1104
- Lam YC, Deng YM, Au CK (2006) A GA/gradient hybrid approach for injection moulding conditions optimisation. *Eng Comput* 21:193–202
- Yuce B, Fruggiero F, Packianather M, Pham D, Mastrocinque E, Lambiasi A, Fera M (2017) Hybrid Genetic Bees Algorithm applied to single machine scheduling with earliness and tardiness penalties. *Comput Ind Eng* 113:842–858
- Vosoughi A, Darabi A (2017) A new hybrid CG-GAs approach for high sensitive optimization problems: with application for parameters estimation of FG nanobeams. *Appl Soft Comput* 52:220–230
- Chen Z-X, Huang H (2011) A hybrid method for intrusion detection with GA-based feature selection. *Intell Autom Soft Comput* 17:175–186
- Mafarja MM, Mirjalili S (2017) Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312
- Taheri K, Hasanipanah M, Golzar SB, Majid MZA (2017) A hybrid artificial bee colony algorithm-artificial neural network for forecasting the blast-produced ground vibration. *Eng Comput* 33:689–700
- Xue Y, Zhong S, Ma T, Cao J (2015) A hybrid evolutionary algorithm for numerical optimization problem. *Intell Autom Soft Comput* 21:473–490
- Singh N, Singh S (2017) A novel hybrid GWO-SCA approach for optimization problems. *Eng Sci Technol Int J* 20:1586–1601
- Pan J-S, Dao T-K, Chu S-C, Nguyen T-T (2018) A novel hybrid GWO-FPA algorithm for optimization applications. *Advances in smart vehicular technology, transportation, communication and applications*. VTCA 2017. Smart innovation, systems and technologies. Springer, Cham, pp 274–281
- Javidrad F, Nazari M (2017) A new hybrid particle swarm and simulated annealing stochastic optimization method. *Appl Soft Comput* 60:634–654
- Kumar N, Vidyarthi DP (2016) A novel hybrid PSO-GA meta-heuristic for scheduling of DAG with communication on multiprocessor systems. *Eng Comput* 32:35–47
- Garg H (2016) A hybrid PSO-GA algorithm for constrained optimization problems. *Appl Math Comput* 274:292–305
- Ali AF, Tawhid MA (2017) A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Eng J* 8:191–206
- Kaur S, Mahajan R (2018) Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egypt Inform J* 19(3):145–150
- Ghasemi E, Kalhori H, Bagherpour R (2016) A new hybrid ANFIS-PSO model for prediction of peak particle velocity due to bench blasting. *Eng Comput* 32:607–614
- Hasanipanah M, Shahnazar A, Bakhshandeh Amnieh H, Jahed Armaghani D (2017) Prediction of air-overpressure caused by mine blasting using a new hybrid PSO-SVR model. *Eng Comput* 33:23–31
- Chopra N, Kumar G, Mehta S (2016) Hybrid GWO-PSO algorithm for solving convex economic load dispatch problem. *Int J Res Adv Technol* 4(6):37–41
- Kamboj VK (2016) A novel hybrid PSO-GWO approach for unit commitment problem. *Neural Comput Appl* 27:1643–1655
- Singh N, Singh SB (2017) Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J Appl Math* 2017:2030489. <https://doi.org/10.1155/2017/2030489>
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Rep* 2005005:2005
- Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 42:627–646
- Dong M, Wang N, Cheng X, Jiang C (2014) Composite differential evolution with modified oracle penalty method for constrained optimization problems. *Math Prob Eng* 2014:617905. <https://doi.org/10.1155/2014/617905>
- Cherri LH, Mundim LR, Andretta M, Toledo FMB, Oliveira JF, Carravilla MA (2016) Robust mixed-integer linear programming models for the irregular strip packing problem. *Eur J Oper Res* 253:570–583
- Alves C, Brás P, Valério de Carvalho J, Pinto T (2012) New constructive algorithms for leather nesting in the automotive industry. *Comput Oper Res* 39(7):1487–1505
- Brás P, Alves C, Valério De Carvalho J, Pinto T (2010) Exploring new constructive algorithms for the leather nesting problem in the automotive industry. *IFAC Proc Vol* 43:225–230
- Crispin A, Clay P, Taylor G, Bayes T, Reedman D (2005) Genetic algorithm coding methods for leather nesting. *Appl Intell* 23:9–20
- Yuping Z, Shouwei J, Chunli Z (2005) A very fast simulated re-annealing algorithm for the leather nesting problem. *Int J Adv Manuf Technol* 25:1113–1118
- Basturk B (2006) An artificial bee colony (abc) algorithm for numeric function optimization. In *IEEE Swarm Intelligence Symposium*. Indianapolis, IN, USA, 2006
- Yu JJ, Li VO (2015) A social spider algorithm for global optimization. *Appl Soft Comput* 30:614–627
- Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proc ICNN'95 Int Conf Neural Netw* 4:1942–1948 IEEE
- Hasanipanah M, Naderi R, Kashir J, Noorani SA, Zeynali Aaq Qaleh A (2017) Prediction of blast-produced ground vibration using particle swarm optimization. *Eng Comput* 33:173–179

42. Ghasemi E (2017) Particle swarm optimization approach for forecasting backbreak induced by bench blasting. *Neural Comput Appl* 28:1855–1862
43. Yagiz S, Ghasemi E, Adoko AC (2018) Prediction of rock brittleness using genetic algorithm and particle swarm optimization techniques. *Geotech Geol Eng* 36:1–11
44. Kaveh A, Zakian P (2017) Improved GWO algorithm for optimal design of truss structures. *Eng Comput* 34:1–23
45. Mirjalili S (2015) How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Appl Intell* 43:150–161
46. Eberhart R, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, vol. 1, pp. 84–88, IEEE
47. Shirani H, Habibi M, Besalatpour A, Esfandiarpour I (2015) Determining the features influencing physical quality of calcareous soils in a semiarid region of Iran using a hybrid PSO-DT algorithm. *Geoderma* 259–260:1–11
48. Li S-F, Cheng C-Y (2017) Particle swarm optimization with fitness adjustment parameters. *Comput Ind Eng* 113:831–841
49. Meng X, Liu Y, Gao X, Zhang H (2014) A new bio-inspired algorithm: chicken swarm optimization. Springer, Cham, pp 86–94
50. James J, Li VO (2015) A social spider algorithm for global optimization. *Appl Soft Comput* 30:614–627
51. James J, Li VO (2015) Parameter sensitivity analysis of social spider algorithm. In: *Evolutionary computation (CEC), 2015 IEEE congress on, IEEE*, pp 3200–3205
52. Dang BT, Vo MC, Truong TK (2017) Social spider algorithm-based spectrum allocation optimization for cognitive radio networks. *Int J Appl Eng Res* 12(13):3879–3887

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.