4 Tornado optimizer-based Coriolis force

In the tornado optimizer with Coriolis force (TOC), it is assumed that there are many windstorms, some thunderstorms, and precipitation phenomena, where tornadoes are generated by windstorms and thunderstorms, and thunderstorms are generated by windstorms. The following are the detailed mathematical models of the proposed TOC optimizer.

4.1 Initialization of population

The proposed TOC optimizer is a population-based algorithm; accordingly, the first step of initiating the optimization process by this optimizer is to randomly create an initial population of designs variables (i.e., windstorms and thunderstorms) between upper bounds (*u*) and lower bounds (*l*). The best individuals (i.e., windstorms and thunderstorms), ranked in terms of having minimum cost function, or in some other cases maximum fitness, are selected to form tornadoes or a tornado if there is only one tornado. A number of good individuals (i.e., values of the cost function close to the current best solution) are chosen as thunderstorms, while all other individuals are called windstorms that eventually evolve into thunderstorms and tornadoes.

To commence TOC as an optimization algorithm, an initial population matrix of *n* individuals (i.e., population size), in a *d*-dimensional search space (i.e., dimension of the problem) is created as a first step. In this, the position of every windstorm, thunderstorm, and tornado indicates a candidate solution to the optimization problem. Equation 21 states how to produce the initial population of windstorms, thunderstorms, and tornadoes in the search domain using a uniform random initialization process.

$$y_{i,j} = l_j + rand \times (u_j - l_j) \tag{18}$$

where *rand* is an arbitrary number generated in the range [0, 1], $y_{i,j}$ is the starting value of the *i*th individual in the *j*th dimension, and u_j and l_j reflect the upper and lower limits of the search space, respectively.

After the creation of *n* individuals, n_{to} individuals are selected from the population that are considered the best candidates to be thunderstorms and tornadoes. Consequently, the individuals with the best values among them are considered tornadoes or are referred to as n_o. Simply put, n_{to} is the summation of the number of thunderstorms and tornadoes, which can be described as exhibited in Eq. 19.

$$n_{to} = n_t + n_o \tag{19}$$

where n_t refers to the number of thunderstorms, while n_o refers to the number of tornadoes, which is equal to one in this work.

The rest of the population forms windstorms. These windstorms may evolve into thunderstorms or they may evolve directly into tornadoes, which can be calculated using Eq. 20.

$$n_w = n - n_{to} \tag{20}$$

where n_w stands for the number of windstorms and n denotes the total population size (i.e., $n = n_w + n_t + n_o$).

The initial population of windstorms, thunderstorms, and tornadoes can be described by a matrix of individuals of size $n \times d$. Therefore, the randomly generated matrix y (that is, the total population) can be shown as follows:

$$y = [y_w, y_t, y_o]_{n\times d} = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,j} & \cdots & y_{1,d} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,j} & \cdots & y_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{i,1} & y_{i,2} & \cdots & y_{i,j} & \cdots & y_{i,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n,1} & y_{n,2} & \cdots & y_{n,j} & \cdots & y_{n,d} \end{bmatrix}_{n\times d} \tag{21}$$

where $y_{i,j}$ denotes the *i*th candidate individual at dimension *j*, which could be a windstorm, a thunderstorm, or a tornado, *d* denotes the number of design variables (i.e., problem dimension), and the components y_w, y_t, and y_o stand for the population of tornadoes, thunderstorms and windstorms, which can be defined as shown in Eqs. 22, 23, and 24, respectively.

$$y_w = \left[y_{w_1}, y_{w_2}, \ldots, y_{w_i}, \ldots, y_{w_{n_w}}\right] \tag{22}$$

$$y_t = \left[y_{t_1}, y_{t_2}, \ldots, y_{t_i}, \ldots, y_{t_{n_t}}\right] \tag{23}$$

$$y_o = \left[y_{o_1}, y_{o_2}, \ldots, y_{o_i}, \ldots, y_{o_{n_o}}\right] \tag{24}$$

where y_{w_i} identifies the *i*th windstorm, y_{t_i} identifies the *i*th thunderstorm, and y_{o_i} represents the *i*th tornado.

As presented in Eq. 21, in a *d* dimensional optimization problem, windstorms, thunderstorms, and tornadoes can be combined together and described by a matrix of appropriate size.

4.2 Fitness evaluation

The cost of the fitness value (i.e., cost function) is computed for each windstorm and thunderstorm by evaluating the cost value as shown below:

$$fit_i = fit(y_{i,1}, y_{i,2}, y_{i,3}, \ldots, y_{i,d}) \tag{25}$$

where fit_i denotes the cost value of the *i*th individual.

Each potential solution to a new windstorm, thunderstorm, or tornado is evaluated based on a fitness criterion created specifically for this purpose. If the newly established position is superior to the present one, the former is then refurbished. As per this, several values of the objective function of the optimization problem of interest are evaluated as a result of putting potential solutions into the decision variables, which can be represented as given in Eq. 26.

$$\vec{fit} = \begin{bmatrix} fit_1 \\ \vdots \\ fit_i \\ \vdots \\ fit_n \end{bmatrix}_{n \times 1} = \begin{bmatrix} fit(y_1) \\ \vdots \\ fit(y_i) \\ \vdots \\ fit(y_n) \end{bmatrix}_{n \times 1} \tag{26}$$

where $\vec{fit}$ is the vector of the acquired fitness function, and fit_i denotes the value of the acquired fitness function on the basis of the *i*th individual.

The value of the fitness function serves as a gauge of the candidate solution's quality in meta-heuristic algorithms like TOC. The population's member that results in the evaluation of the best value for the fitness function is referred to as the best population's member. This member is updated in each iteration loop of the proposed optimizer because the candidate solutions are updated throughout. In the simulation of the proposed optimizer, individuals stay in their locations if they are better than the new locations.

4.3 Evolution of windstorms

Windstorms tend to move toward tornadoes and thunderstorms based on the volume and intensity of their evolution. This means that windstorms evolve into tornadoes more often than thunderstorms.

4.3.1 Initialization of windstorms' population

As described above, n_w windstorms are generated, such that this number of candidate individuals are selected from the entire population. Equation 27 shows y_w (i.e., population of windstorms) that evolve into tornadoes or thunderstorms. Indeed, Eq. 27 is part of Eq. 21 (i.e., all individuals in the population):

$$\begin{aligned} y_w &= \begin{bmatrix} y_{w_1} & y_{w_2} & \cdots & y_{w_i} & \cdots y_{w_{n_w}} \end{bmatrix} \\ &= \begin{bmatrix} y_{w_{1,1}} & y_{w_{1,2}} & \cdots & y_{w_{1,d}} \\ y_{w_{2,1}} & y_{w_{2,2}} & \cdots & y_{w_{2,d}} \\ \vdots & \vdots & \vdots & \vdots \\ y_{w_{n_w,1}} & y_{w_{n_w,2}} & \cdots & y_{w_{n_w,d}} \end{bmatrix}_{n_w \times d} \end{aligned} \tag{27}$$

4.3.2 Formation of windstorms

Depending on the size and power of the evolution of windstorms, tornadoes and each thunderstorm ingests windstorms. In any manner, one of the best ways to distribute windstorms between tornadoes and thunderstorms in a proportional way is to use cost functions (fitness functions) for tornadoes and thunderstorms. Hence, the number of windstorms that accede into thunderstorms and/or tornadoes mutates. The designated windstorms for tornadoes and each thunderstorm are evaluated using the following mathematical formulas:

$$f_k = fit_k - fit_{n_{to}+1} \tag{28}$$

where $k = 1, 2, 3, \ldots, n_{to}$, and f_k specifies the cost value of the *k*th thunderstorm associated with a tornado.

$$n_{\dot{w}_k} = \left\lfloor \left\{ \left| \frac{f_k}{\sum_{k=1}^{n_{to}} f_k} \right| \times n_w \right\} \right\rceil \tag{29}$$

where $\lfloor \cdot \rceil$ stands for the round operator, $k = 1, 2, \ldots, n_{to}$, and $n_{\dot{w}_k}$ is the number of windstorms that evolve or assign into specified thunderstorms or tornadoes.

In fact, in the implementation of the proposed optimizer, the costs of tornadoes and each thunderstorm have been deducted by the cost of an individual (i.e., $n_{to} + 1$) in the population of windstorms (see Eq. 27) as can be seen in Eq. 28. Based on their strength and rate of growth, windstorms frequently develop into thunderstorms and tornadoes. It implies more windstorms evolve into tornadoes than into thunderstorms. Hence, one of the finest techniques to distribute windstorms among tornadoes and thunderstorms in a proportionate manner is to employ objective criteria (fitness functions) for tornadoes and thunderstorms.

With the use of Eqs. 28 and 29, the best solution (i.e., tornadoes) will be able to control and retain more windstorms. It is worth noting that windstorms will be randomly selected from the population of windstorms. Each windstorm is controlled by one of the best individuals (i.e., tornados or thunderstorms). Thus, windstorms cannot be assigned to more than one best individual. However, in rare situations, the sum of $n_{\dot{w}_k}$ in Eq. 29 may not equal n_w. This issue has been sorted out in the implementation code of TOC. In this case, the number of windstorms deemed for thunderstorms and tornadoes is randomly decreased or increased by subtracting or adding a single value (i.e., ± 1). Thus, the total number of windstorms assigned to thunderstorms and tornadoes will be exactly equal to n_w.

The speed and direction of movement of windstorms may be affected by the Coriolis force, as shown below:

4.4 Windstorm velocity with the Coriolis force

For large-scale atmospheric turbulences that do not require the windstorm to move in a straight line, there is a three-way balance among the Coriolis force, centrifugal force, and the pressure gradient forces. As per this, the gradient windstorm speed that forms thunderstorms and tornadoes can be identified as shown in Eq. 30.

$$\vec{v}_i^{t+1} = \begin{cases} \eta(\mu \vec{v}_i^t - c\frac{(f \times R_l)}{2} + \sqrt{CF_l}) & rand \geq 0.5 \\ \eta(\mu \vec{v}_i^t - c\frac{(f \times R_r)}{2} + \sqrt{CF_r}) & rand < 0.5 \end{cases} \tag{30}$$

where $i = 1, 2, \ldots, n_w$, is the windstorm's index for a population of size n_w, $\vec{v}_i^{t+1}$ denotes the new velocity vector of the *i*th windstorm, $\vec{v}_i^t$ defines the current speed vector of the *i*th windstorm, *rand* refers to a generated random number with uniform distribution in the scope [0, 1], η identifies a shrinkage factor presented to simulate the convergence conduct of windstorms as defined in Eq. 31, μ implements the fuzzy adaptive kinetic energy of windstorms defined as exposed in Eq. 32, R_l is the radius of curvature of the trajectory of windstorms in the Northern Hemisphere defined as given by Eq. 33, R_r is the radius of curvature of the path of windstorms in the Southern Hemisphere given by Eq. 34, *c* stands for a created random number in different ranges defined as shown in Eq. 35, and *f*, CF_l, and CF_r can be defined as presented in Eqs. 38, 39, and 40, respectively.

In Eq. 30, $rand \geq 0.5$ demonstrates that the motion of the windstorms is in the northern hemisphere, and $rand < 0.5$ illustrates that the motion of windstorms is in the southern hemisphere. Thus, *rand* was used to simulate the motion of windstorms between the Northern and Southern Hemispheres

$$\eta = \frac{2}{\left|2 - \chi - \sqrt{\chi^2 - 4\chi}\right|} \tag{31}$$

where χ identifies the rate of acceleration of windstorms, which is equal to 4.10, where this value was obtained after careful investigation.

Equation 31 was introduced in the proposed optimizer as a constriction factor for ameliorating the convergence behavior. The constriction factor η in this equation has a value of 0.7298. Mathematically, the constriction factor is analogous to momentum energy, which can be important to provide windstorms with the necessary power to reach the target to form thunderstorms and tornadoes. This factor can be essential for the success of the proposed optimizer and achieving promising performance levels.

Beyond the constriction factor η in Eq. 31, a fuzzy adaptive μ was applied in the proposed optimizer with a random version setting of what is defined in Eq. 32.

$$\mu = 0.5 + \frac{rand}{2} \tag{32}$$

where *rand* denotes a generated random number with uniform distribution in the scope [0, 1].

Equation 32 was used to give a fuzzy adaptive random number for dynamic system optimization, where this random μ has an expectation of 0.75.

$$R_l = \frac{2}{1 + e^{(-t+T/2)/2}} \tag{33}$$

$$R_r = \frac{-2}{1 + e^{(-t+T/2)/2}} \tag{34}$$

where *t* and *T* stand for the current and maximum number of iteration indices, respectively.

In fact, windstorm speeds can exhibit both clockwise and counterclockwise rotation. Most (but not all) tornadoes in the northern hemisphere rotate counterclockwise, because they develop from large, rotating supercell thunderstorms.

$$c = b_r \times \delta_1 \times w_r \tag{35}$$

where b_r is a constant equal to 100000, δ_1 stands for a change in the sign presented as shown in Eq. 36, and w_r identifies a random value generated with different ranges defined as given by Eq. 37.

$$\delta_1 = f_d \lceil 2 \times rand \rceil \tag{36}$$

where f_d represents a function of values 1 and -1 to represent the change in sign.

$$w_r = \frac{2 \times rand - (rand + rand)}{w_{min} + rand \times (w_{max} - w_{min})} \tag{37}$$

where *rand* stands for generated random values in the range]0, 1], and w_{min} and w_{max} are fixed values equal to 1.0 and 4.0, respectively.

$$f = 2 \cdot \Omega \times \sin(-1 + 2 \cdot rand) \tag{38}$$

where Ω stands for the angular rotation rate that is equal to 0.7292115E-04 radians s^{-1} , *rand* stands for a random number generated with uniform distribution in the range [0, 1] Vallis (2017), and $-1 + 2 \cdot rand$ specifies a random value for the latitude.

$$CF_l = \frac{(f^2 \times R_l^2)}{4} - R_l \times \phi_i^t \tag{39}$$

where ϕ_i^t is the component of the pressure gradient force (PGF) normal to the direction of the current *i*th windstorm at the specified *t* iteration as exposed in Eq. 41.

$$CF_r = \frac{(f^2 \times R_r^2)}{4} - R_r \times \phi_i^t \tag{40}$$

$$\phi_i^t = y_{o_\zeta}^t - y_{w_i}^t \tag{41}$$

where $y_{o_\zeta}^t$ is the current position vector of the tornado at a random index ζ at the *t*th iteration, $y_{w_i}^t$ identifies a position vector of a windstorm at the *t*th iteration, and ζ is a random index of a tornado defined as shown in Eq. 42.

$$\zeta = \lceil n_o \times rand\,(1, n_o) \rceil \tag{42}$$

where $rand(1, n_o)$ implements a uniformly generated vector of random values with a uniform distribution in the interval $[0, 1]$.

Equation 30 is subject to the constraints given in Eqs. 43, 44, and 45.

$$\phi_i^t = \begin{cases} -\phi_i^t & \mathrm{sgn}(R_l) \geq 0 \ \ \& \ \ \mathrm{sgn}(\phi_i^t) \geq 0 \\ -\phi_i^t & \mathrm{sgn}(R_r) \leq 0 \ \ \& \ \ \mathrm{sgn}(\phi_i^t) \leq 0 \\ \phi_i^t & otherwise \end{cases} \tag{43}$$

$$CF_l^t = \begin{cases} -CF_l^t & \mathrm{sgn}(CF_l^t) < 0 \ \ \& \ \ rand \geq 0.5 \\ CF_l^t & otherwise \end{cases} \tag{44}$$

$$CF_r^t = \begin{cases} -CF_r^t & \mathrm{sgn}(CF_r^t) < 0 \ \ \& \ \ rand < 0.5 \\ CF_r^t & otherwise \end{cases} \tag{45}$$

where *rand* stands for a random number generated with uniform distribution in the range [0, 1].

4.4.1 Evolution of windstorms to tornadoes

The process of evolution of windstorms into tornadoes is performed in the TOC optimizer. A tornado or tornadoes are formed from windstorms or thunderstorms when windstorms evolve to tornadoes or thunderstorms. This evolution process can be simulated mathematically as shown in Eq. 46.

$$\vec{y}_{w_i}^{t+1} = \vec{y}_{w_i}^{t} + 2 \times \alpha \times (\vec{y}_{o_i}^{t} - rand_w) + \vec{v}_i^{t+1} \tag{46}$$

where $\vec{y}_{w_i}^{t+1}$ and $\vec{y}_{w_i}^{t}$ define the next and current position vectors of the *i*th windstorm at iterations $(t+1)$ and *t*, respectively, $\vec{y}_{o_i}^{t}$ defines the current position vector of the *i*th tornado at iteration *t*, $(\vec{y}_{o_i}^{t} - rand_w)$ denotes the difference between the evolution of windstorms into tornadoes and the random formation of windstorms, the components $rand_w$ and α stand for random values that can be defined as presented in Eqs. 47 and 48, respectively.

$$rand_w = \vec{y}_{w_i}\left(\lfloor n_w \times rand(1, n_w)\rfloor + 1\right) \tag{47}$$

where $rand_w$ is an index vector for randomly selected windstorms.

$$\alpha = |2a_y \cdot rand - rand| \tag{48}$$

where *rand* denotes a random value created with a uniform distribution in the range [0, 1], and a_y represents an exponential parameter defined as shown in Eq. 49.

$$a_y = \frac{(T - (t^{a_0}/T))}{T} \tag{49}$$

where a_0 denotes a constant value of 2.0 and was found after extensive analysis.

Equation 46 can essentially be thought of as an update formula for new positions of windstorms that evolve into tornadoes.

4.4.2 Evolution of windstorms to thunderstorms

As noted above, there are *n* individuals of which n_t is selected as thunderstorms and n_o is selected as tornadoes. In this work, we assume that there is only one tornado. A schematic view of a windstorm evolving into a particular thunderstorm along its contact line is seen in Fig. 4.

The distance γ between windstorms and thunderstorm may be amended randomly as given in Eq. 50.

$$\gamma \in (0, \rho \times x), \quad \rho > 0.5 \tag{50}$$

where x is the present separation between windstorms and thunderstorms, $0.5 < \rho < 2$ where 2 may be the optimal value of ρ, and γ conforms to a random number between 0 and $\rho \times x$ that is uniformly distributed or chosen from a plausible distribution.

Windstorms can evolve in several directions approaching thunderstorms when $\rho > 0.5$ is set. This idea may be utilized as well to explain how thunderstorms may evolve into tornadoes. In essence, to consummate the exploration and exploitation phases in TOC, the evolution process of windstorms into thunderstorms may be simulated as follows:

$$\begin{aligned} \vec{y}^{t+1}_{w_{j+\sum_1^{n_{\dot{w}_k}}}} =& \vec{y}^{t}_{w_{j+\sum_1^{n_{\dot{w}_k}}}} \\ &+2 \times rand \times (\vec{y}^{t}_{t_i} - \vec{y}^{t}_{w_{j+\sum_1^{n_{\dot{w}_k}}}}) \\ &+2 \times rand \times (\vec{y}^{t}_{o_i} - \vec{y}^{t}_{w_{j+\sum_1^{n_{\dot{w}_k}}}}) \end{aligned} \tag{51}$$

where $\vec{y}^{t+1}_{w_{j+\sum_1^{n_{\dot{w}_k}}}}$ and $\vec{y}^{t}_{w_{j+\sum_1^{n_{\dot{w}_k}}}}$ represent the next and current position vectors of windstorms developing into thunderstorms at iterations $(t+1)$ and *t*, respectively, $\vec{y}^{t}_{t_i}$ represents the current position vector of the *i*th thunderstorm at iteration *t*, and *rand* stands for a random number produced between 0 and 1 with uniform distribution.

Equation 51 is regarded as a mathematical formula for new positions of windstorms that evolve into thunderstorms.

4.5 Evolution of thunderstorms to tornadoes

In the exploration and exploitation phases of the proposed optimizer, the new position of thunderstorms during evolving into tornadoes can be simulated in the manner defined below:

$$\begin{aligned} \vec{y}^{t+1}_{t_i} =& \vec{y}^{t}_{t_i} + 2 \times \alpha \times (\vec{y}^{t}_{t_i} - y^{t}_{o_\varsigma}) \\ &+2 \times \alpha \times (\vec{y}^{t}_{t_{\vec{p}}} - \vec{y}^{t}_{t_i}) \end{aligned} \tag{52}$$

where $\vec{y}^{t+1}_{t_i}$ and $\vec{y}^{t}_{t_i}$ represent the next and current position vectors of thunderstorms developing into tornadoes at iterations $(t+1)$ and *t*, respectively, $y^{t}_{o_\varsigma}$ identifies a position vector for a tornado at a random index ζ, and $\vec{y}^{t}_{t_{\vec{p}}}$ identifies a position vector for a thunderstorm

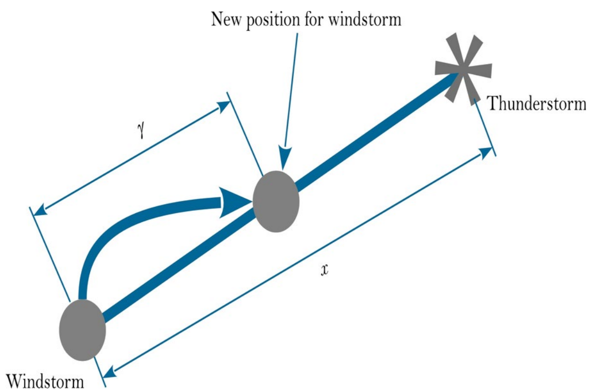

Fig. 4 A diagram showing how a specific thunderstorm develops from a windstorm (a thunderstorm is represented by a asterisk, while a windstorm is represented by a ball)

at a random index vector $\bar{p}$ which is the index vector for randomly selected thunderstorms identified as shown in Eq. 53.

$$\vec{p} = \lfloor n_t \cdot rand\,(1, n_t) + 1 \rfloor \tag{53}$$

where *rand* is a uniformly distributed random number in the range of [0, 1].

Equation 52 is the updated mathematical model for thunderstorms that evolve into tornadoes. Notations marked with a vector sign correspond to vector values, otherwise the rest of the notations and parameters are scalar values.

The positions of the thunderstorms and windstorms are exchanged if the windstorm's solution is better than that of its connected thunderstorm (i.e., the windstorm becomes a thunderstorm, and the thunderstorm becomes a windstorm). As a result, windstorms from preceding thunderstorms serve as new thunderstorms and are better windstorms (in terms of the cost function value). In fact, the current thunderstorm is in charge of all earlier windstorms. The transition between a thunderstorm and a tornado, as well as between windstorms and tornadoes, may be done similarly. In this scenario, the evolving thunderstorm will behave like a new tornado, and the outgoing tornado will be a new thunderstorm with its own windstorms that can be pushed exactly in its direction. Therefore, the windstorms connected to the prior thunderstorms-which are now a new tornado-will behave as windstorms that are directly developing into the new tornado. The interchange of windstorms and thunderstorms in the population of the proposed optimizer can be observed as shown in Fig. 5.

Figure 6 (which also incorporates the concept from Fig. 4) shows how the optimization process of the proposed optimizers can be evolved, with balls, asterisks, and the bullet represent windstorms, thunderstorms, and tornadoes, respectively. The white (empty) shapes show where the windstorms and thunderstorms have moved to.

4.6 Random formation of windstorms

The stochastic formation of windstorms process is defined in the proposed TOC-based optimization to enhance its exploration capability. To be more precise, the random formation of windstorms enables TOC to avoid falling into local solutions and immature convergence. Basically, windstorms evolve in random locations when they evolve into thunderstorms or evolve into tornadoes, resulting in mature tornadoes in different positions. This process is applied to both windstorms and thunderstorms, which must be checked to see if they are close enough to a tornado to make this process occur. For this purpose, the following mathematical formula can be utilized to accomplish the random process of forming windstorms into tornadoes:

$$\begin{aligned} \vec{y}_{w_i}^{t+1} &= \vec{y}_{w_i}^{t} - (2 \times a_y \times (rand \times (l - u) - l)) \times \delta_2 \\ &\left\| \vec{y}_{w_i}^{t} - \vec{y}_{o_i}^{t} \right\| < \nu \end{aligned} \tag{54}$$

where l and u refer to the bottom and top limits of the search area, respectively, *rand* is a random number systematically inserted into the range [0, 1], δ_2 stands for a change in the sign specified as given in Eq. 55, $\|\cdot\|$ refers to a norm operator, and ν is an exponential function defined as shown in Eq. 56, which is capable of generating small numbers.

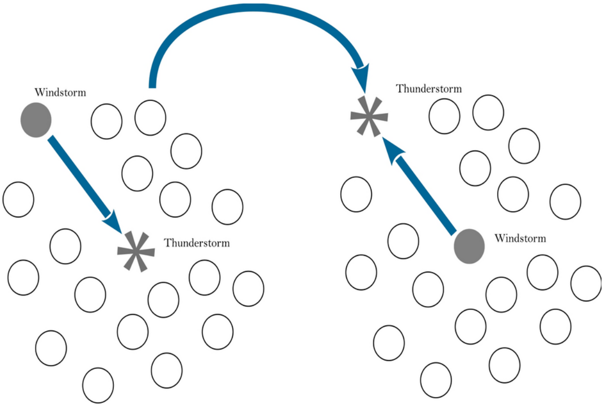

Fig. 5 The placements of windstorms and thunderstorms are switched, with the black ball displaying the best windstorms among the others, and the asterisk representing thunderstorms

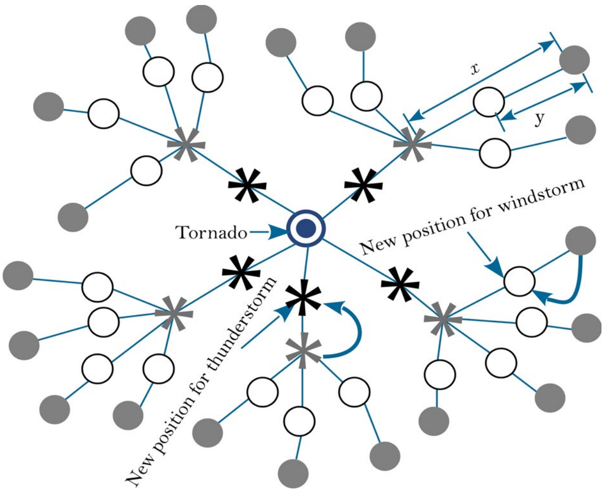

Fig. 6 Illustration of the strategies used in the proposed optimizer, where the bullet represents a tornado

$$\delta_2 = f_d \lfloor 2 \times rand + 1 \rfloor \tag{55}$$

where f_d represents a function of the values 1 and -1 to represent the change in sign.

$$\nu = \left(0.1e^{\left(-0.1(t/T)^{0.1}\right)}\right)^{16} \tag{56}$$

where t represents the current iteration index and T represents the maximum iteration index.

Continuing the random formation of windstorm, the following mathematical formula can be utilized to accomplish the random process of windstorms formation into thunderstorms:

$$\begin{aligned} \vec{y}^{\,t+1}_{w_{j+\sum_{i=1}^{n_{\dot{w}_k}}}} =& \vec{y}^{\,t}_{w_{j+\sum_{i=1}^{n_{\dot{w}_k}}}} \\ & -(2 \times a_y \times (rand \times (l-u) - l)) \times \delta_2 \\ & \left\| \vec{y}^{\,t}_{w_i} - \vec{y}^{\,t}_{t_i} \right\| < \nu \end{aligned} \tag{57}$$

Accordingly, Eqs. 54 and 57 were introduced in this work to specify the new locations of newly formed windstorms. As presented in Eq. 56, ν regulates the search intensity close to the tornado. In view of this, big values of ν may discourage further searches, but small values may promote search activity in the immediate vicinity of the tornado.

From a mathematical perspective, the parameter a_y creates an adaptive function over the iteration loops of TOC. Using this adaptive function, windstorms with a_y following the condition utilized in Eqs. 54 and 57 are scattered about it. In fact, with the use of this method, TOC may execute a better search surrounding the tornado during its exploitation phase.

Additionally, as observed in nature, certain thunderstorms form slowly since just a few windstorms give rise to them. As a result, they will not be able to go closer to a tornado and may eventually get smaller after making certain motions. The TOC optimizer adopts the parameter a_y to boost the random evolution process of windstorms to reinforce this idea. Then, using Eqs 54 and 57, new windstorms will be produced at new locations, equal to the number of prior windstorms and thunderstorms. As can be seen from Eq. 52, thunderstorms are not regarded as fixed points in the proposed TOC and must evolve towards tornadoes (i.e., the optimal solution). This process (developing windstorms into thunderstorms and then thunderstorms into tornadoes) promotes indirect development in the direction of the best solution. In short, as the iterations of the TOC optimizer continue, the likelihood of random generation process of windstorms decreases.

4.7 Complexity analysis

A function that relates the dimension and input size of a given input problem to the time of execution of the optimization algorithm under investigation may be employed to quantify the computational complexity of the algorithm. This basically singles out how the complexity issue of the proposed optimizer can be studied. The time and space computational complexities of the developed optimizer are described below in terms of Big-O notation as a standard expression.

4.7.1 Time complexity

Time complexity of the proposed optimizer can be generally represented using Big-O notation as follows.

$$\begin{aligned}\mathcal{O}(TOC) =& \mathcal{O}(prob.\ def.) + \mathcal{O}(pop.\ init.) \\ &+ \mathcal{O}(cost\ fun.) + \mathcal{O}(sol.\ update)\end{aligned} \tag{58}$$

The computational complexities of the proposed TOC optimizer depend on the time complications of several components connected to the relevant problem and the proposed method, as shown in Eq. 58. There are various temporal complexity issues with each of these components, where these elements can be described as follows:

1. Defining the problem takes $\mathcal{O}(1)$ time.
2. The time required for population initialization is $\mathcal{O}(v \times n \times d)$ time.
3. The time required for cost function evaluation is $\mathcal{O}(v \times K \times c \times n)$ time, where c represents the cost of the criterion assessment.

4. Updates to the solution and their evaluation take $\mathcal{O}(v \times K \times n \times d)$ time.The time complexity of the optimization process is dependent on a number of variables, including v, n, d, T, and c, where these parameters stand for the total amount of evaluations, the amount of individuals, the overall dimensions of the optimization problem, the amount of iteration steps, and the cost of the fitness criterion, respectively. This is explained above and in connection with what is presented in Eq. 58. One may describe the total time complexity of TOC in distinct components as shown in Eq. 59 by considering the points above and Eq. 58.

$$\mathcal{O}(TOC) = \mathcal{O}(v(1 + nd + Kcn + 5Knd)) \tag{59}$$

As $1 \ll Kcn$, $1 \ll 5Knd$, $nd \ll Kcn$, $nd \ll 5Knd$, and $Knd < 5Knd$, Eq. 59 may be streamlined to what is presented in Eq. 60:

$$\mathcal{O}(TOC) \cong O(Kcn + Knd) \tag{60}$$

As it turns out, the time complexity issue of TOC in terms of Big-O notation is of the polynomial order. The proposed TOC optimizer might be seen in this sense as a computationally efficient optimization technique. The number of decision variables in the problem (d), the cost of the problem's objective criterion (c), the number of individuals (n), and the overall amount of iterations (T) can all be expressed as the main considerations to recognize the computational complexity of TOC when addressing an optimization problem.

4.7.2 Space complexity

The parameters of the number of windstorms, thunderstorms, and tornadoes, and the size of the problem of interest influence the space complexity of TOC in terms of the available memory space. This reveals how much room TOC would take up while starting the optimization process. Accordingly, the spatial complexity of TOC may be well represented as exhibited in Eq. 61.

$$\mathcal{O}(nd) \tag{61}$$

4.8 Implementation steps of the proposed optimizer

The general picture of the evolution process of tornadoes and the stochastic formation methods of windstorms and thunderstorms have led us to develop mathematical models of the proposed TOC optimizer and the implementation of optimization. While solving optimization problems, the optimization process of TOC endeavors to advance in the direction of the global optimal solution. This is because it is very probable that windstorms and thunderstorms as well as tornadoes will turn out and spirally rotate in the encircling area in the search space to locate a better solution. The implementation of this capacity relies on where the optimal windstorms, thunderstorms, and tornadoes are found. Consequently, windstorms and thunderstorms are consistently capable of turning all around the potential areas in the search space to evolve into tornadoes. The key procedural steps mentioned in Algorithm 1 summarize the pseudo code of the proposed optimizer.

Define and initialize the parameter settings of TOC.
$t \leftarrow$ Iteration counter
$T \leftarrow$ Maximum number of iterations
$n \leftarrow$ Population size
$n_w \leftarrow$ Number of windstorms
$n_t \leftarrow$ Number of thunderstorms
Create a random initial population using Eq. 18.
Evaluate the position of the initial population using Eq. 26.
while $(t \leq T)$ **do**
  Update the adaptive parameters using the respective formulas
  for i=1 to n_w **do**
    if $(rand \geq 0.5)$ **then**
      $\vec{v}_i^{t+1} = \eta(\mu \vec{v}_i^t - c\frac{(f \times R_l)}{2} + \sqrt{CF_l})$
    else
      $\vec{v}_i^{t+1} = \eta(\mu \vec{v}_i^t - c\frac{(f \times R_r)}{2} + \sqrt{CF_r})$
    end if
  end for
  for i=1 to n_w **do**
    $\vec{y}_{w_i}^{t+1} = \vec{y}_{w_i}^t + 2 \times \alpha \times (\vec{y}_{o_i}^t - rand_w) + \vec{v}_i^{t+1}$
  end for
  Investigate and update the feasibility of windstorms' positions
  for i=1 to n_t **do**
    for j=1 to $n_{\dot{w}_i}$ **do**
      $\vec{y}^{t+1}_{w_{j+\Sigma_1^{n_{\dot{w}_k}}}} = \vec{y}^t_{w_{j+\Sigma_1^{n_{\dot{w}_k}}}} + 2 \times rand \times (\vec{y}_{t_i}^t - \vec{y}^t_{w_{j+\Sigma_1^{n_{\dot{w}_k}}}}) + 2 \times rand \times (\vec{y}_{o_i}^t - \vec{y}^t_{w_{j+\Sigma_1^{n_{\dot{w}_k}}}})$
    end for
  end for
  Investigate and update the feasibility of windstorms' positions
  for i=1 to n_t **do**
    $\vec{y}_{t_i}^{t+1} = \vec{y}_{t_i}^t + 2 \times \alpha \times (\vec{y}_{t_i}^t - y_{o_\varsigma}^t) + 2 \times \alpha \times (\vec{y}^t_{t_{\vec{p}}} - \vec{y}_{t_i}^t)$
  end for
  Investigate and update the feasibility of thunderstorms' positions
  for i=1 to n_w **do**
    if $\left\| \vec{y}_{w_i}^t - \vec{y}_{o_i}^t \right\| < \nu$ **then**
      $\vec{y}_{w_i}^{t+1} = \vec{y}_{w_i}^t - (2 \times a_y \times (rand \times (l - u) - l)) \times \delta_2$
    end if
  end for
  for i=1 to n_t **do**
    if $\left\| \vec{y}_{w_i}^t - \vec{y}_{t_i}^t \right\| < \nu$ **then**
      for j=1 to $n_{\dot{w}_i}$ **do**
        $\vec{y}^{t+1}_{w_{j+\Sigma_{i=1}^{n_{\dot{w}_k}}}} = \vec{y}^t_{w_{j+\Sigma_{i=1}^{n_{\dot{w}_k}}}} - (2 \times a_y \times (rand \times (l - u) - l)) \times \delta_2$
      end for
    end if
  end for
  $t = t + 1$
  Retain the best solutions
end while

Algorithm 1 A pseudo code summing up the iterative steps of the proposed TOC optimizer.

The proposed TOC optimizer starts the optimization process by randomly generating the locations of the population of windstorms, thunderstorms, and tornadoes in the search space, in accordance with the stages of the pseudo code presented in Algorithm 1. To update these individuals' positions during each function evaluation, TOC uses Eqs. 30, 46, 51, 52, 54, and 57. According to the simulated stages of the proposed optimizer, if any of the entities (i.e., windstorms, thunderstorms, and tornadoes) depart the search space, they will all be brought back. In each function evaluation, the solutions are evaluated using a predetermined fitness criterion, and the individuals with the best fitness values are identified by updating the fitness function. It is indicated that the best position for individuals to achieve their goals is the most appropriate solution. In each function evaluation, all algorithmic steps-aside from the initialization ones-are repeated until the predetermined overall amount of function evaluations has been attained. The proposed models of this optimizer are continuously capable of spiraling and spinning out in all places in the search space, according to the theoretical claims of the optimizer described above.

4.9 Characteristics of TOC

The proposed TOC optimizer as a nature-inspired meta-heuristic has two capabilities during the optimization of a particular problem inside the search space: exploration and exploitation. The convergence of the proposed TOC optimizer towards the global optimal solution makes these capabilities possible. Specifically, convergence occurs when the majority of thunderstorms and windstorms congregate in the same area of the search space. TOC uses a number of crucial parameters that promote the exploration and exploitation aspects including μ, α, a_y, and ν which are defined in Eqs. 32, 48, 49, and 56, respectively. By adjusting these parameters, the proposed TOC optimizer may more effectively explore the search space for every potential solution to find the sub-optimal or optimal solutions. Therefore, these control parameters are useful in TOC to implement promising convergence property. As windstorms and thunderstorms develop into tornadoes in the TOC optimizer, these search agents-windstorms, thunderstorms, and tornadoes-can update their positions in accordance with the mathematical models and tuning criteria of TOC implemented by the evolution models of the search agents. The models of TOC are presented in Eqs. 30, 46, 51, 52, 54, and 57. Windstorms and thunderstorms are assumed to efficiently evolve into tornadoes within the search space in each of the presented models. Additionally, the random motions of windstorms and thunderstorms occur as a result of their random evolution, which forces them to move to random locations. Thus, tornadoes, thunderstorms, and windstorms all explore the search space in various directions and places, implying that additional attractive areas could hold better solutions. In summary, TOC offers a few characteristics based on its fundamental idea, which may be summed up as follows: (1) The position update models of the proposed TOC optimizer efficiently help the population search agents explore and exploit each region of the search space, (2) The random motions of windstorms and their random formation into tornadoes in the search space using Eqs. 46, 54 and 57 boost population diversity of TOC and guarantee a reasonable convergence rate, demonstrating an effective trade-off between exploration and exploitation, (3) The random movements of windstorms and their evolution into thunderstorms using Eq. 51 and the random motions of thunderstorms and their evolution into tornadoes in the search space using Eq. 52 increase the diversity of the population, ensure sensible convergence property, and provide an efficient equilibrium between exploration and exploitation aspects, (4) The number of parameters in TOC is reasonably acceptable, and these are promising operators to provide a high level of performance.

5 Comparative analysis of TOC with other optimizers

This section compares the TOC method with various well-established meta-heuristic algorithms, such as particle swarm optimization (PSO), genetic algorithms (GAs), differential evolution (DE), and ant colony optimization (ACO) algorithm.

5.1 Particle swarm optimization

PSO imitates the collective cooperative social behavior of living organisms, such as flocks of birds, fish, and many other species of creatures (Kennedy and Eberhart 1995). Artificial