Computational Intelligence
AN INTERNATIONAL JOURNAL

Computational Intelligence, Volume 32, Number 2, 2016

CRYSTAL ENERGY OPTIMIZATION ALGORITHM

XIANG FENG, MEIYI MA, AND HUIQUN YU

Department of Computer Science and Engineering, East China University of Science and Technology, China

Nature has always been a muse for those who dream in art or science. As it goes, optimization algorithms inspired by nature have been widely used to solve various scientific and engineering problems because of their intelligence and simplicity. As a novel nature-inspired algorithm, the crystal energy optimizer (CEO) is proposed in this article. The proposed CEO is motivated by the following general observation on lake freezing in nature: the dynamics of crystals have possession of parallelism, openness, local interactivity, and self-organization. It stimulates us to extend a crystal dynamic model in physics to a generalized crystal energy optimizer for traveling salesman problems, so as to exploit the advantages of crystal dynamic system and to realize the aforementioned purposes. The proposed CEO has these advantages: (1) it has the ability to perform large-scale distributed parallel optimization; (2) it can converge and avoid local optimum; and (3) it is flexible and easy to adapt to a wide range of optimization problems.

Received 29 December 2013; Revised 4 July 2014; Accepted 12 July 2014

Key words: crystal energy optimizer (CEO), computational intelligence, parallel algorithm, nature-inspired algorithm, traveling salesman problem (TSP).

1. INTRODUCTION

1.1. Background

When the Nobel Prize in Chemistry was awarded for research inspired by jellyfish, a comment from *Nature* goes, "The natural world serves scientists as a constant signpost towards phenomena yet unknown. The natural world continues to hint at solutions to modern technological dilemmas, and that when it comes to simple and effective solutions, nature is usually well ahead of man" (Anonymous 2008).

Nature is enlightening and encouraging human beings to seek inspiration from it. The tradition of biologically inspired computing extends back more than half a century to the original musings of Alan Turing about artificial intelligence and John von Neumann's (1955) early work on self-replicating cellular automata in the 1940s. Since then, computer scientists have frequently turned to biological processes for inspiration. In Jeffrey O. Kephart's (2011) *Learning from Nature*, he agrees that "Indeed, the influence of biological analogies is attested by sub-fields of computer science, such as artificial neural networks, genetic algorithms, and evolutionary computation."

Methods mimicking and modifying nature-inspired models have been proposed in recent years. As a result, plenty of unsolved and time-consuming problems in real life have been solved by them effectively. However, existing algorithms could not always satisfy the increasingly overwhelming requirement of super large and complicated instances in the real world. The incremental demand of intelligent and efficient algorithms attracts more researches to exploit computational intelligence in nature.

By this circumstance, obeying the rules of being "simple and effective," a novel nature-inspired algorithm, the crystal energy optimizer (CEO), is proposed in this article. Just

Address correspondence to Xiang Feng and Meiyi Ma, Department of Computer Science and Engineering, East China University of Science and Technology, Meilong Road 130, Shanghai 200237, China; e-mail: xfeng@ecust.edu.cn, mameiyi0106@163.com.

like the common "jellyfish" in the sea, lake freezing in winter is one of the normal but enlightening phenomena in nature. As we all may notice, with the temperature falling in winter, the lake begins to precipitate little ice crystals, which will link into bigger crystal sheets. At last, the whole surface of the lake freezes like a mirror, as shown in Figure 1. From water to ice, behaviors of crystals in the lake are driven by the energy transfer, which is caused by the change of temperature. Building on this process, models concerning energy transfer among crystals are established as the fundamental of the proposed CEO, which is expected to be a parallel algorithm with high intelligence and efficiency, especially for large-scale problems.

1.2. Related Work

Laws of nature keep all creatures born, growing, and living in their environment harmoniously. Human beings draw inspiration from them to design strategies and propose approaches. Most of the strategies and approaches drawn from natural intelligence can be categorized into three classes. First, evolutionary algorithms such as genetic algorithm (Forrest 1993) are inspired by the Darwinian theory. Second, behavior-related and swarm approaches (Mehrjoo, Sarrafzadeh, and Mehrjoo 2015), such as the ant colony optimizer (ACO) (Bonabeau, Dorigo, and Theraulaz 2000), particle swarm optimizer (PSO) Zhan et al. (2011), and group search optimizer He, Wu, and Saunders (2009), are inspired by the social intelligence of animals. Third, phenomenon-mimicked models, for example, the root growth algorithm Zhang, Zhu, and Chen (2014), hurricane-based optimization algorithm (Rbouh and Imrani 2014), and artificial plant optimization algorithm Cui et al. (2012), are inspired by nature rules. Most well-known and well-developed intelligent algorithms existing belong to the first two categories. Although there are plenty of enlightening phenomena in nature, only a few related approaches are proposed. The third category remains to be exploited. For example, the freezing phenomenon of a lake is a promising instance with both mechanisms of swarm intelligence and phenomenal mimicry. From the perspective of the lake, all the crystals inside will precipitate and grow under the influence of energy transfer (Maybank and Barthakur 1967). From the perspective of crystal individuals, they perform their behaviors independently. Thus, the inherent merits of both swarm intelligence and natural rules could be represented by the crystal models at the same time.

Although outstanding results are reached by the intelligent methods discussed earlier, a local optimum is inevitable sometimes for the inherent deficiency of mathematical models

FIGURE 1. The frozen lake.

(Lim and Zhu 2013). This is mainly because some formulas and parameters in them are obtained through a series of experiments without supporting theoretical evidence. However, physics-inspired approaches can avoid this problem. Models built on the theories in physics, such as Newtonian mechanics and principle of conservation of energy, usually have strong support, as well as more stable and convincing results. For example, the simulated annealing algorithm (Kirkpatrick, Gelatt, and Vecchi 1983) is built on the fundamental of thermodynamics. It has a superior ability to avoid local optima. Different from other nature-inspired algorithms, the proposed CEO algorithm is influenced by the energy transfer functions, which obey the theorem of dynamics. Furthermore, the mathematical formulas are supported by the dynamic functions of crystallization in *nature*.

Another problem of most of the existing intelligent algorithms is their frequent interactions between individuals or agents. Two consequences could be brought. First, if one individual or agent falls into a local optimum, it is easy for others to be stuck in local as well (Ishteva et al. 2011). Second, these strong interactions require the algorithms to be executed step by step (Ling et al. 2011; Sun et al. 2013). Therefore, much computer time will be wasted, especially when the scale of a problem is large. However, as we have discussed earlier, the crystals in the proposed CEO are independent. Few interactions among them not only reduce the possibility of local optimum but also make the proposed CEO an outstanding property of parallelism. Another effective way to keep the proposed CEO from the local optimum is the wind-blow model, where crystals could be dropped off and relinked at a better position.

The intelligent algorithms have been applied to solve numbers of complex problems (Liu and Zhou 2014), such as the series of continuous benchmark functions (He et al. 2009), wireless sensor networks (He et al. 2012; Sadi and Ergen 2013), Altman information security (Altman et al. 2013; Tang et al. 2013), and so on (Lim and Mat Isa 2014). The performance on these applications is promising. However, most of these algorithms are designed particularly for one or two types of problems; thus, their scope of application is limited. For example, notwithstanding the superior results of PSO in continuous functions (Schmitt and Wanka 2015), its performance on discrete problems is not desirable even with the help of Sigmoid functions and 0–1 models. Also, the performance of the group search optimizer in dynamic problems (Teimourzadeh and Zare 2014) is not competitive with static functions. Not particularly designed for a certain type of problem, the proposed CEO has a much broader application than the other intelligent algorithms for three reasons. First, the CEO can be used to solve either continuous or discrete problems. Second, the status of crystals is only affected by the energy transfer, setting the crystals free from the constraints of the geographic position. Therefore, the CEO could be applied in both static and dynamic fields. Third, the parallelism of CEO makes it capable of solving large-scale problems in a relatively short time, while many other similar algorithms even could not find an acceptable solution.

1.3. Motivation

The proposed CEO is motivated by the following general observation on lake freezing in nature: the dynamics of crystals have possession of parallelism, openness, local interactivity, and self-organization. It stimulates us to extend a crystal dynamic model in physics to a generalized CEO for traveling salesman problems (TSPs), so as to exploit the advantages of crystal dynamic system and to realize the aforementioned purposes. The advantages of the proposed CEO are highlighted as follows.

(1) Building on the intelligent and strongly supported mathematical model, the CEO algorithm is expected to have a promising optimization performance. Because the

models of CEO have the characteristics of three nature-inspired models as we have discussed earlier, they are supposed to perform their combination merits. First, from the perspective of the phenomena mimicry model, the CEO models follow the intelligent rules of being "simple and effective." Moreover, CEO is a branch of swarm intelligence, and thus, the decentralized and self-organized system could give it the ability to obtain independence and global optimization. At last, the accuracy and effectiveness of mathematical models could be guaranteed by the physics-inspired model.

(2) Local optima are supposed to be prevented validly by the proposed CEO according to the following protective measures. Because crystals in the CEO precipitate and grow individually by their energy transfer functions, there are few interactions existing among them. Without "copying" or "following" behaviors, the probability of all the agents falling into a local optimum declines. Even if some crystals are stuck in the suboptimum, by the effect of wind blowing, their conditions could be changed by the wind.

(3) By its inherent parallel structure, the CEO model admits a highly parallel algorithmic implementation and thus possesses the ability to deal with large-scale complex problems. With few interactions, the energy of crystals in CEO could be calculated separately on different computing cells. Freezing on parallel could reduce the computer time, which is inversely proportional to the number of processors. Thus, the high efficiency of CEO could be reached, making CEO suitable for large-scale problems.

(4) The application of CEO could have a broader perspective. To start with, instead of designing for some certain type of problems, the procedure of CEO is relatively simple and suitable for both continuous and discrete problems. In addition, the energy of CEO is mainly affected by the environment and updated in each iteration, making it less relevant with the position of the crystal. Therefore, it is capable of solving both static and dynamic problems. In the end, the ability of solving large-scale problems, as we have analyzed, also extends its scale of application to a profound field, where only a few algorithms could reach.

1.4. Contribution

The main contribution of this article is as follows.

(1) Inspired by the nature phenomenon of freezing and crystal behaviors, the precipitation–growth model and the wind-blow model are built physically. Furthermore, according to the theory of dynamics and functions of energy transfer, mathematical models are established. The precipitation–growth model can find the suboptimal solution in a very short time. Then, the wind-blow model will lead it to the optimum.

(2) Based on the aforementioned models, the CEO is proposed. It is a parallel algorithm with high efficiency, especially for large-scale problems.

(3) The effectiveness and convergence of CEO are analyzed theoretically. Through the theory of Lyapunov stability and the convergence theorem, the convergence of energy of CEO is proved. Therefore, the applicability not only for the TSP problem but also for all the optimization problems is verified.

(4) The effectiveness, efficiency, and parallelism of the proposed approach are tested by a number of simulations. First, by comparing the quality of solution and computer time before and after wind blowing, we have two conclusions. On the one hand, an acceptable suboptimal solution for some applications can be obtained by the CEO in a very short time. On the other hand, the wind-blowing process can prevent solution from the local optimum effectively. Second, by comparing the quality of the solution of the CEO

with that of other algorithms, the superior performance of the CEO is illustrated. Third, the high efficiency and less time consumed for the large-scale instances of the CEO illustrate its effectiveness on large-scale problems. Last, its property of parallelism is verified by parallel experiments.

1.5. Organization

The rest of this article is organized as follows. Crystal energy models are built in Section 2 mathematically. The parallel CEO is proposed based on crystal energy models in Section 3, and its solution for TSP is introduced in Section 4. In addition, the theoretical foundation and convergence analysis of our proposed algorithm are discussed in Section 5. Furthermore, experimental results of CEO are analyzed and compared in Section 6. In Section 7, we draw the conclusions and discuss about future works.

2. CRYSTAL ENERGY OPTIMIZER

2.1. Description for Freezing Behaviors of Crystal

There is a descent of temperature in winter. When it falls down below freezing point, lakes in the north will begin to freeze. The freezing phenomenon of a lake is a gradual transition of water from liquid to solid, which is caused by the transfer of its energy. Equation (1) from Murray, Knopf, and Bertram (2005) indicates the relationship between temperature and energy, where $\Delta E(Q)$ is the variation of energy, m is the quality of entity, C is the specific heat capacity, and ΔT is the variation of temperature. The main process of freezing is as follows: (a) the temperature descent in the environment makes the lake release energy to the environment; (b) according to equation (1), the temperature of the lake falls down; and (c) when the temperature of the lake reaches the freezing point (temperature threshold of freezing), crystals begin to precipitate and link together.

$$\Delta \mathrm{T} = \Delta E(Q)/mC. \tag{1}$$

The transfer of energy during this process is so complicated that a number of factors should be taken into consideration, such as the temperature of the lake and environment, properties of the lake, composition of the atmosphere, wind and pressure around them, and inter-transfer energy between them.

However, we simplify the freezing behaviors of crystals by proposing the following assumptions.

Hypothesis 1 (Process). Releasing energy to the atmosphere, crystals in the lake begin to precipitate when their energy arrives at the threshold. Then, precipitated ones will join together through crystal links. By which means, a crystal shell is formed. With more and more crystals precipitating, the crystal shell grows until the whole lake is frozen.

Hypothesis 2 (Shell). It is easier for the crystal near the shore of the lake to precipitate, because the water there is shallow and far away from the center of the lake. Therefore, there is a crystal shell formed on the edge of the lake at the beginning of the freezing. The order of freezing should be from the edge to the center gradually.

Hypothesis 3 (Wind). Wind blowing over the lake has two major functions, taking away the energy and breaking up the crystal links on the lake.

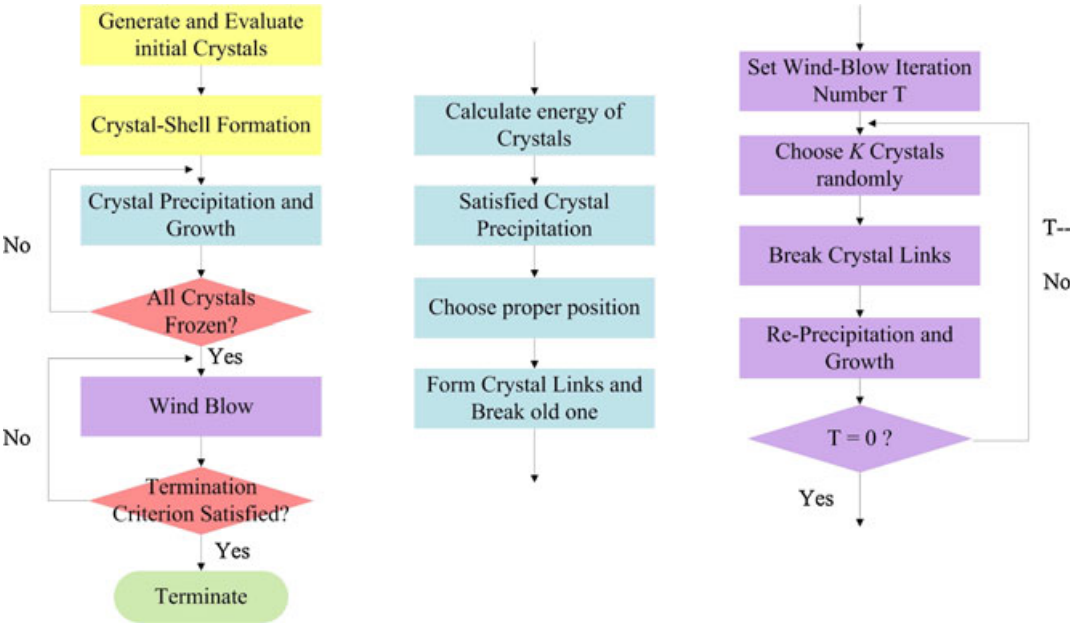
Hypothesis 4 (Energy). During the process of freezing, there are three major forms of energy affecting the energy of lake. The first one is the energy of the center of the lake, where the temperature is highest. It can be regarded as the source of heat for the lake. Therefore, when the crystals lose energy, they will recharge from the center. The second effect comes from the precipitated crystals. They will try their best to "attract" crystals to join them by absorbing their energy and forming the crystal links. The last major influence is the wind, which has been discussed in Hypothesis 3.

Inspired by the phenomenon of freezing of a lake (assumptions made earlier), crystal models are built based on the freezing behaviors of crystals. The flow charts of the models of CEO are shown in Figure 2, where Figure 2(a) is the overall flow of the CEO including several submodels. Meanwhile, Figure 2(b and c) shows the specific processes of precipitation–growth and wind-blow, respectively. In addition, mathematical models are further developed elaborately in the following subsections.

2.2. Formation

According to Hypotheses 1 and 2, crystals on the edge of the lake will precipitate at the beginning of the formation. It looks like a convex hull when all the crystals by the shore precipitate and link together. Therefore, the initial crystal shell is formed.

Considering the different statues of crystals, sets are defined first to clarify their relationships and transition. According to Definitions 1 and 2, a set involved in the freezing process is defined as $Crystal - \varepsilon$, which consists of two subsets $CrystalW - \varepsilon$ and $CrystalI - \varepsilon$. The later one is the precipitated ice crystal, and the former one is the water crystal. Another important set is $Shell - \varepsilon$, which is also a subset of $Crystal - \varepsilon$, consisting of the crystals on the edge.



(a) Overall Flowchart (b) Precipitation-Growth (c) Wind-Blow

FIGURE 2. The flowchart of crystal energy optimizer models.

Definition 1. Supposing that $Crystal - \varepsilon = \{c_1, \ldots, c_i\}$ represents all the crystals in the lake, including the frozen and unfrozen parts, and ordered collection $CrystalI - \varepsilon = \{c_1, \ldots, c_i\}$ represents the crystal precipitated, we have

$$Crystal - \varepsilon = CrystalI - \varepsilon + CrystalW - \varepsilon. \tag{2}$$

Definition 2. Supposing that $Shell - \varepsilon = \{s_1, \ldots, s_m\}$ represents the crystal shell; thus,

$$Shell - \varepsilon \subset CrystalI - \varepsilon. \tag{3}$$

Definition 3. The two-dimensional matrix $R = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \cdots & r_{ij} & \cdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix}$ represents the crystal links, and $r_{ij} = \{0, 1\}$ represents the link between crystals c_i and c_j. If $r_{ij} = 1$, there is a link between c_i and c_j. Otherwise, if $r_{ij} = 0$, there is no link between c_i and c_j.

The two-dimensional matrix $D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \cdots & d_{ij} & \cdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix}$ represents the distance between crystals, where d_{ij} is the distance between crystal c_i and c_j.

The total length L of links is donated by $L = R \circ D$, where "$\circ$" means $\sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij} * d_{ij}$, $r_{ij} \in R, d_{ij} \in D$, i.e.,

$$L = R \circ D = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij} * d_{ij}. \tag{4}$$

2.3. Precipitation–Growth

After the formation of the crystal shell, other crystals keep precipitating and growing because of the decrease of energy. According to Hypothesis 4, the influence on energy of crystals comes from three roles, the lake center, precipitated crystals, and wind. As shown in Figure 3, for any crystal c_i, it absorbs energy from the center and releases energy to the ice crystal. At the same time, the energy of all the crystals is taken away by the wind. The energy of crystals is defined in Definitions 4–8.

Definition 4. For any crystal c_i, E_{i0} is its initial energy. At time t, if $Q_i(t)$ is the energy influence from the environment, the total energy of c_i at time t is

$$E_i(t) = E_{i0} + Q_i(t). \tag{5}$$

Supposing that there are n crystals in the lake, the total energy of the lake at time t is

$$E(t) = \sum_{i=0}^{n} E_i(t) = \sum_{i=0}^{n} (E_{i0} + Q_i(t)). \tag{6}$$

Particularly, the influence from the environment $Q_i(t)$ consists of three parts as defined in Definition 5.

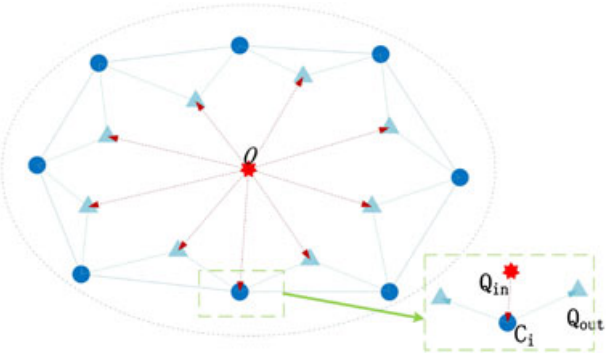


FIGURE 3. The energy transfer on a lake.

Definition 5. Supposing that $P_i(t)$, $S_i(t)$, and $W_i(t)$ are the effects from the center energy of the lake, frozen crystal energy, and wind, respectively, their weighted sum gives the total influence as

$$Q_i(t) = \lambda_1 P_i(t) + \lambda_2 S_i(t) + \lambda_3 W_i(t) \tag{7}$$

where λ_1, λ_2, and λ_3 are the decision factors with two functions. On the one hand, their signs indicate the direction of influence. For instance, λ_1 is positive, meaning that the crystal absorbs energy from the center. Meanwhile, λ_2 and λ_3 are negative, indicating that both ice crystals and wind take away their energy. On the other hand, the values of these decision factors vary in different stages of freezing. The reason is that different factors dominate different processes of freezing, which will help to speed up the convergence of the algorithm.

During the process of precipitation, the energy of crystals is taken away by the wind and resupplied by the lake center. By this means, the center and wind are the dominant factors of freezing. Thus, we have $\lambda_1 > \lambda_2$ and $\lambda_3 > \lambda_2$.

After precipitating, crystals link together because of the influence from the ice crystals. By this circumstance, ice crystals are the dominant factors. Therefore, the relationship turns to $\lambda_2 > \lambda_1$ and $\lambda_3 > \lambda_1$.

Definition 6. Supposing that the location of crystal c_i is L_i; then the distance d_{ij} between crystals c_i and c_j is given by

$$d_{ij} = ||(L_i - L_j)||. \tag{8}$$

Therefore, the distance from crystal c_i to the center d_{i0} is given by

$$d_{i0} = ||(L_i - L_0)||. \tag{9}$$

According to McKay et al. (1985), the balance of upward conduction of heat at a given depth in the ice cover gives

$$Z = \frac{k(T_0 - T_s) - S_0(1-a)(1-r)h}{v\rho l}.$$

In this equation, the meanings of parameters are as follows. Z is the equilibrium thickness of the ice cover; k is the thermal conductivity of ice; T_0 is the temperature of the ice–water interface; T_S is the yearly averaged temperature of the surface; S_0 is the solar

radiation incident on the lake surface, and the exponential term gives attenuation with depth into the ice with an extinction path length of h; a is the albedo of the lake; r is the fraction of the lake; the heat flow caused by the latent heat release at the ice–water interface is defined as $L = v\rho l$, where v is the rate of formation of new ice averaged over the entire year, ρ is the density of the ice, and l is the latent heat of water.

By analyzing this equation and combining it with the assumptions we have made in our models, it is simplified and transformed as follows.

First, according to equation (1), the relationship between energy and temperature, the variation of temperature ΔT, can be expressed as

$$\Delta T = T_0 - T_s = \frac{\Delta E}{mC}.$$

Second, CEO models are built on the phenomena of freezing of the surface of the lake. Therefore, the depth of the lake is not taken into consideration. Thus, we assume the thickness of frozen lake as $Z = 1$ and the extinction depth as $h = 0$. Then we have

$$1 = \frac{k\frac{\Delta E}{mC}}{v\rho l}.$$

Thus, the changing of energy of ice on the lake caused by the heat of the lake is

$$\Delta E = \frac{v\rho mC}{k} \cdot l.$$

Third, for a given crystal c_i, v, ρ, m, C, and k are constant parameters. To simplify the expression, we have

$$\alpha = \frac{v\rho mC}{k}.$$

However, the heat of the water is not the same for all the crystals. As we have assumed, the source of heat from the lake is the center of the lake, so its energy effect is inversely proportional to the distance from the crystal to the center:

$$l = \frac{\mathbb{P}}{d_{i0}}$$

where constant parameter $\mathbb{P}$ is the energy released by the center.

In sum, the influence on crystal c_i from the center of lake at time t, $P_i(t)$, equals the variation energy from the center heat to the crystal ΔE, which can be expressed as equation (10) in Definition 7.

Definition 7. The energy crystal c_i absorbed from the center of lake $P_i(t)$ at time t is inversely proportional to its distance from the center:

$$P_i(t) = \frac{\alpha\mathbb{P}}{d_{i0}} = \frac{\alpha\mathbb{P}}{||L_i - L_0||} \tag{10}$$

where constant parameter $\mathbb{P}$ is the energy released by the center and α is a scale factor.

Krinner et al. (2004) show that the enhanced ice sheet growth in Eurasia is due to several adjacent ice-dammed lakes in a large area. From a micro-perspective, we infer that crystals that precipitated in a lake also have an influence on other crystals. This inference also agrees with the theory of heat conduction (Strunk, Tracy, and Kleiber 1973), which indicates that there are three equations for heat transfer in different conditions. For the heat transferred between crystals, we have

$$dQ/dt = kA/d(T - T_0)$$

where dQ/dt is the total rate of heat flow, k is the thermal conductivity, A is the heat transfer area, d is the thickness over which a temperature gradient exists, and $T - T_0$ is the difference of temperature.

We also simplify and transform this equation according to our models and assumptions as follows.

First of all, based on equation (1), the difference of temperature can be expressed as the difference of energy, as

$$\Delta E = mC(T - T_0).$$

Second, for any crystal c_i, the values of the total rate of heat flow, the thermal conductivity, and the heat transfer area are the same. Therefore, dQ/dt, k, and A are constant.

Third, the thickness over which a temperature gradient exists d is different because the distances between two crystals are different.

Therefore, the equation of energy transfer $S_{ji}(t)$ between two crystals c_i and c_j is transformed into

$$\Delta E_{ij} = \frac{kAmC}{v} \cdot \frac{1}{d_{ij}}.$$

$\beta \cdot \mathbb{S}$ is used to substitute the constant part, as

$$\beta \cdot \mathbb{S} = \frac{kAmC}{v}$$

where β is a scale factor and $\mathbb{S}$ is the energy of the crystal source.

Thus, we have the energy function of ice crystals $E_{ji}(t)$ as

$$E_{ij}(t) = \beta \mathbb{S} \cdot \frac{1}{d_{ij}}.$$

In our CEO models, a crystal just precipitated c_i will join between two ice crystals. Therefore, two adjacent crystals that have the most effect on c_i will break the existing link and link with c_i. This process is defined by equation (11) in Definition 8. Their effect is defined by equation (12) in Definition 8.

Definition 8. Supposing that crystal c_i only releases energy to the closest two adjacent ice crystals, its released energy at time t, $S_i(t)$, is

$$S_i(t) = \max_{j \in [0,N]} [S_{ji}(t)] \tag{11}$$

$$S_{ji}(t) = E_{ji}(t) + E_{(j+1)i}(t) - E_{j(j+1)}(t) = \beta\mathbb{S} \cdot \left(\frac{1}{d_{ij}} + \frac{1}{d_{i(j+1)}} - \frac{1}{d_{j(j+1)}}\right) \quad (12)$$

where $S_{ji}(t)$ is the influence by frozen crystal c_j on its neighbor crystal c_{j+1}.

2.4. Behaviors of the Wind

As maintained by Bell (2008), when the wind (air) blows over the surface of the water, huge quantities of ice will be moved around by melting the ice sheets. It indicates two major behaviors of wind (air flow) blowing over a lake:

- Taking away the energy of crystals
- Breaking up the existing links between crystals

The first behavior, as defined in Definition 9, is one of the main factors inducing the precipitation and growth of crystals. It will keep taking away the energy of crystals constantly.

Definition 9. For any crystal c_i, the energy taken away by the wind at time t, $W_i(t)$, is constant, which is defined by

$$W_i(t) = \mathbb{C} \quad (13)$$

where $\mathbb{C}$ is constant.

The second behavior will break the existing links of crystals.

According to Maybank and Barthakur (1967), the ice crystals are more numerous, thinner, and more fragile with the decrease of temperature. Crystals will shatter one after another during the course of freezing. Then a crystal link will be allowed to resume growth and produce the fragile dendritic filaments.

The growth rates dm/dt are defined as

$$\frac{dm}{dt} \approx 4\pi C \frac{\sigma}{f(T)}$$

where C is a shape factor that assumes different values depending on the geometry of the crystal, σ is the supersaturation of the environment relative to ice, and $f(T)$ is a function of temperature only. Because C and σ are the same for all the crystals, temperature is the only factor affecting the crystal growth rate. It also can be expressed by energy according to equation (1). When wind blows over the surface of the lake, energy will be taken away, and the temperature will keep falling. With the temperature going down after all the crystal precipitated, the crystal shell will begin to "shatter" and "resume" by the effect of the wind. Therefore, the shatter-and-resume crystals are defined in Definition 10.

Definition 10. Assume that K crystals are chosen to melt and re-precipitate randomly at one time and repeat this process T times; thus, the set of shatter-and-resume crystals C_{drop} is

$$C_{drop} = \sum_{j=1}^{T} C_{drop}^{j}$$

$$C_{drop}^{j} = \{c_1, \ldots, c_i, \ldots, c_K\}$$

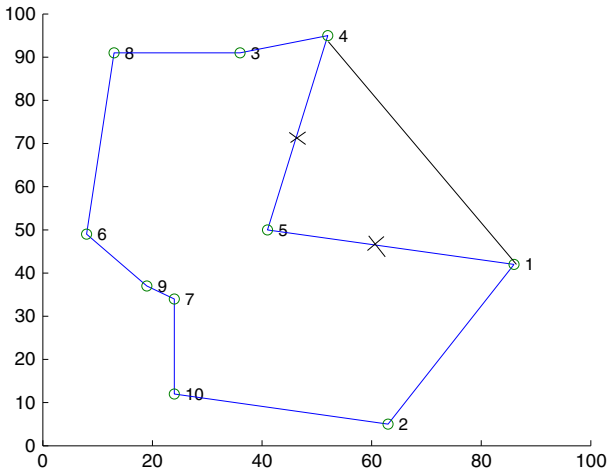


FIGURE 4. The wind-blow model.

where crystal c_i is obtained by

$$\forall c_i \in Crystal - \varepsilon$$

$$c_i \otimes c_{i+1}, c_i \otimes c_{i-1} \quad \text{and} \quad c_{i+1} \oplus c_{i-1}. \tag{14}$$

In Definition 10, $\otimes$ represents cutting off the link between crystals c_i and c_{i+1}, while $\oplus$ represents rebuilding the connection. As shown in Figure 4, there are 10 crystals completely frozen. When wind blows, r_{54} and r_{51} are broken to release crystal c_5. Furthermore, with their energy reducing, it will rejoin the crystal shell again on the position that attracts it most.

During the process of freezing, some crystals may precipitate earlier than other crystals near them, by which circumstances it will link to the ice crystals that are relatively far. It is not the optimal way of linking. Then the CEO may prematurely find a local optimal solution. However, the break–reconnection process by the wind-blow model can change this situation. When some links break up, these premature crystals may break from the crystal shell and reconnect to a better position. Therefore, it can prevent the algorithm from falling into the local optimum effectively.

Therefore, these crystals have a second chance to link to a better place by the wind-blow model. The wind-blow model changes the connection structure of crystals and prevents the algorithm from falling into the local optimum effectively.

3. THE PARALLEL CRYSTAL ENERGY ALGORITHM

Based on the models established in Section 2, the CEO algorithm is proposed in this section. The process of calculation is introduced in Algorithm 1. The main flow of the CEO can be described as follows:

Step 1: The crystal shell forms.
Step 2: Crystals with low energy precipitate and link to the crystal shell.
Step 3: After all the crystals are precipitated, K crystals break up links and drop off randomly.

Step 4: These crystals re-precipitate and link to the crystal shell as in step 2.
Step 5: Step 4 is repeated for T iterations, and final links (i.e., the solution) are obtained.

Algorithm 1 The parallel crystal energy algorithm

Require: Positions of crystals on the lake
Initial energy of crystals E_{i0} and energy threshold E_T
Scale factors α, β, and γ
Decision factors λ_1, λ_2, and λ_3
Wind-blowing iteration number T and crystal number K
Ensure: The links among crystals and their energy
1 Initialize N crystals and parameters —*Parallel*
2 Form the crystal shell
3 **While** $N > 0$ **do**
Calculate the energy of crystals according to equations (5)–(13) —*Parallel*
If the energy of crystal c_i reaches E_T —*Parallel*
Crystal c_i precipitates —*Parallel*
Calculate the influence from ice crystals on c_i according to equation (12) —*Parallel*
Crystal c_i links to the crystal shell by forming two links —*Parallel*
End If
N - -
End While
4 **While** $T > 0$ **do**
Choose K crystals randomly and drop them out of the crystal shell as in equation (14) —*Parallel*
Re-precipitate and relink crystals to the crystal shell as in step 3 —*Parallel*
T - -
End While
5 Output the final links

The CEO algorithm proposed in this article is a parallel algorithm. The parallel structure of the CEO algorithm is shown in Figure 5, where $Crystal - \varepsilon$ represents the set of all the crystals. As shown in Figure 5, computing cells are allocated according to the structure of the CEO. This is the distribution-maximized condition, where each crystal corresponds to a computing cell C_i. After all the crystals of a computing cell finish precipitating and growing, their computing cell is released. Compared with single computing, parallel computing of the CEO algorithm not only increases the computing efficiency but also releases the excess resources occupied by the algorithm to save space overhead.

It is profoundly efficient that each crystal is calculated in parallel in the CEO algorithm. It provides us a flexible interface to calculate more complicated energy functions effectively. The effectiveness of parallelism is more obvious in large-scale problems.

4. THE CEO FOR TSP

4.1. Description for TSP

The mathematical description for TSP is as follows. If the order of traversal for cities (City $= \{city_1, city_2, \ldots, city_n\}$) is $S = \{s_1, s_2, \ldots, s_n\}$, where $s_i \in V(i = 1, 2, \ldots, n)$ and $s_{n+1} = s_1$, then the object is

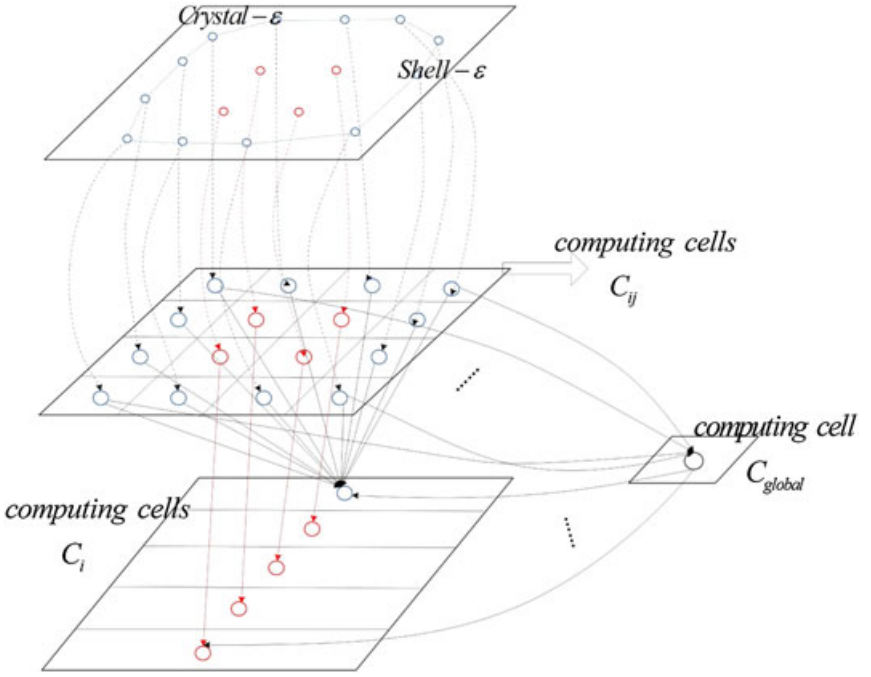


FIGURE 5. The parallel calculation structure of the crystal energy optimizer.

$$\min_{S\in\Omega} \sum_{i=1}^{n} d_{s_t s_{t+1}} \tag{15}$$

$$d_{ij} = \sqrt{(z_i - z_j)^2 + (y_i - y_j)^2} \tag{16}$$

where Ω is the set of all the possible paths between n cities, d_{ij} is the path between $city_i$ and $city_j$, and x_{ij} represents the reachability between city $city_i$ and $city_j$, i.e.,

$$x_{ij} = \begin{cases} 0, & d_{ij} = \infty \\ 1, & d_{ij} \text{ exist} \end{cases}. \tag{17}$$

4.2. Solutions

In this subsection, we will explain the detailed steps of CEO solving the TSP by setting a simple example with 10 cities. The cities in the TSP are represented by the crystals in the CEO. The final links between crystals are the paths of optimal solution for TSP by the end of iterations. Therefore, an ordered set of crystals at the end is the traverse order of cites. The detailed elements of CEO corresponding to TSP are shown in Table 1. The flow of the solution will be introduced step by step as follows. The data of the process are shown in Table 2, and the solution process is in Figure 6. The energy has been highlighted when the statue of crystal changes.

Example 1. A salesman is going to visit 10 cities with the least length of paths. The positions of cities (x, y) are shown in Table 2.

TABLE 1. Solutions of Crystal Energy Optimizer (CEO) on the Traveling Salesman Problem (TSP).

CEO	TSP
Lake	The area with all the cities
Crystal c_i	$city_i$ to visit
Link between crystals c_i and c_j	Path between $city_i$ and $city_j$
Ordered set $\{c_i\}$	Traverse order

Step 1: To find the optimal way, 10 crystals are initialized at the positions of cities with original energy $E_{0i} = 10{,}000$ in the range of cities (100 * 100).
Step 2: The crystal shell is formed by linking c_1, c_2, c_3, c_4, c_5, and c_7, as shown in Figure 6(a). The energy of these crystals decreases to 0.
Step 3: According to equations (7), (10), and (13), the energy of un-precipitated crystals is calculated. α is a random number with a range of [0, 1], $\mathbb{P} = E_{01} = 10{,}000$, and $\mathbb{C} = E_{01}/N = 1{,}000$.

For example, when $t = 1$,

$$P_6(1) = E_{i6} - \mathbb{C} + \alpha\mathbb{P}/d_{60}$$
$$P_6(1) = 10{,}000 - 1{,}000 + 0.5 * 10{,}000/9487 = 9{,}527.048109.$$

Step 4: After a few iterations, when the energy of some crystal(s) is below 0, it/they will precipitate. It/they will link to the crystal shell according to equations (11) and (12).

For example, when $t = 11$, $E_{i9} = -163.6$; thus, c_9 precipitates.

The energy influence from the crystal shell is calculated as equation (12). β is a random number with a range of [0, 1].

$$\mathbb{S} = E_{i0} = 10{,}000.$$

The energy from c_5 and its neighbors c_7–c_9 is calculated as

$$S_{59}(11) = \beta\mathbb{S}(1/d_{59} + 1/d_{79} - 1/d_{57}) = 540.05.$$

Thus, we have the energy influence of different crystals:

$$S_{59}(11) = 540.05, S_{19}(11) = 511.65, S_{29}(11) = 62.43$$

$$S_{79}(11) = 199.03, S_{49}(11) = 33.88, S_{39}(11) = -82.8.$$

Therefore, $\max\{S_{j9}\} = S_{59}$; thus, c_9 should join between c_5 and c_7. $Link_{57}$ is broken, and $Link_{59}$ and $Link_{79}$ are formed, as shown in Figure 6(b).
Step 5: Repeat steps 3 and 4; crystals precipitate and link to the shell. In this case, c_8, c_6, and c_{10} precipitate at $t = 12$ and $t = 13$.

At the end of $t = 13$, the initial formation of the crystal shell is formed, as shown in Figure 6(c). The total length of links, also known as the length of paths, is 276.73.
Step 6: When all the crystals have precipitated, the wind blows over the lake. K crystals are chosen to drop off the shell. For instance, when $K = 3$, c_6, c_8, and c_9 are chosen. Repeat steps 3–5; three crystals will re-precipitate and relink to the shell with the order of c_8, c_9, and c_6. Repeat this process T times.

TABLE 2. Energy of Crystals during Iteration.

	Position		Precipitation–growth energy								Wind-blow energy			
Crystal	x	y	0	1	2	⋯	10	11	12	13	14	15	16	17
1	79	61	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
2	12	60	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
3	98	45	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
4	85	4	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
5	6	52	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
6	47	41	10,000	9,527.0	8,843.2	⋯	2,108.1	1,319.0	635.2	**−259.3**	**2,000**	1,687.2	863.1	**−512.9**
7	33	11	10,000	**0**	0	⋯	0	0	0	0	0	0	0	0
8	64	46	10,000	9,343.4	8,549.4	⋯	1,373.6	510.9	**−282.9**	0	**2,000**	**−92.1**	0	0
9	24	46	10,000	9,190.1	8,304.1	⋯	760.3	**−163.7**	0	0	**2,000**	963.8	**−231.5**	0
10	58	56	10,000	9,500	8,800	⋯	2,000	1,200	500	**−400**	0	0	0	0

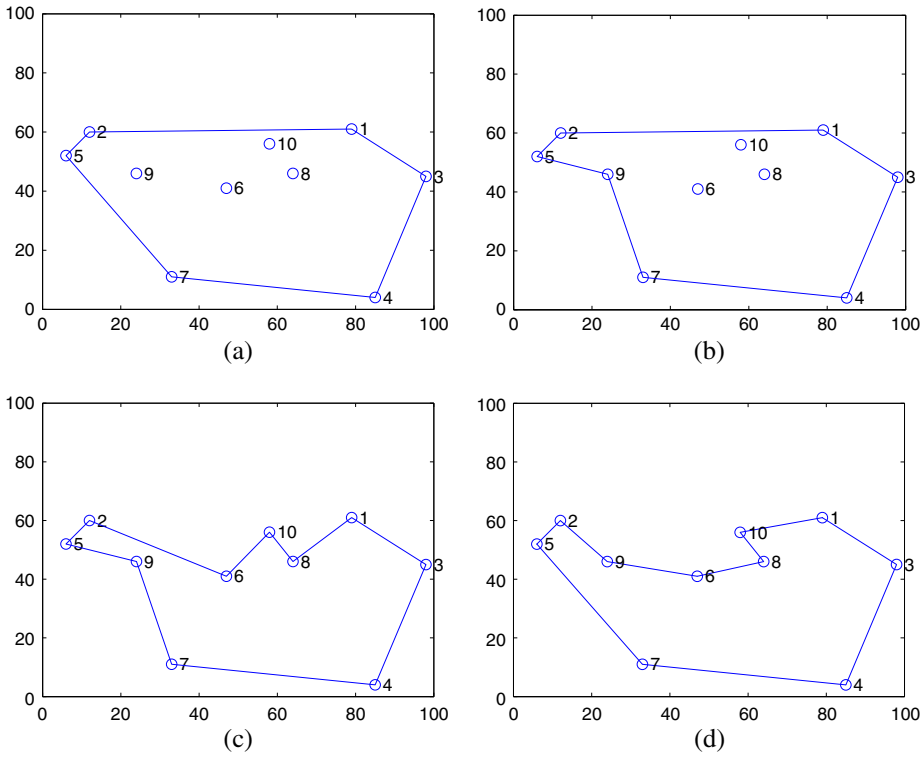


FIGURE 6. An example of the crystal energy optimizer on the traveling salesman problem with 10 cities.

The paths after wind blowing are shown in Figure 6(d). The total length of links is 272.35, which is better than the former solution before wind blowing.

5. THE THEORETICAL FOUNDATION OF CEO

5.1. Convergence proofs

The convergence of the algorithm is verified mathematically from two aspects.

(1) The total length of crystal links.
The length function of links is used to evaluate the effectiveness of CEO. If it is verified that the total length of links decreases to a minimum, the convergence of the algorithm is verified as well.
(2) The total energy of crystals.
The crystal precipitates and grows because of the reduction of the energy, during which process the total energy of crystals decreases monotonically to a minimum.

Lemma 1. The optimal way to build a crystal shell is by forming a convex hull. That is, the length of the convex-hull crystal shell is less than any other method

$$\forall link, d_{convex} \geq d_{others}$$

where d_{convex} and d_{others} represent the length of the crystal shell and other methods, respectively.

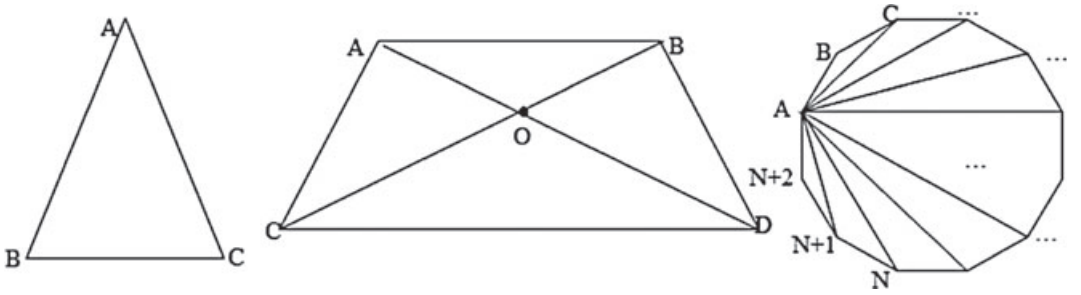


FIGURE 7. Crystal shells with three crystals, four crystals, and $N + 2$ crystals, respectively.

Proof. Mathematical induction:

Assume that the number of edges for a convex polygon is n; thus, we have the following:

(1) If $n = 3$, there is only one way to link three points, the triangle ΔABC is as shown in Figure 7(a). There is no doubt that the convex hull is the minimal way to link.
(2) If $n = 4$, there are two ways to link, as shown in Figure 7(b), one of which is the convex hull $(A- > B- > D- > C- > A)$, and the other one is the cross way with order $(A- > B- > C- > D- > A)$ or $(A- > C- > B- > D- > A)$.

The total length of the crystal links is calculated by

$$\begin{aligned} \mathrm{D}_1 &= d_{AB} + d_{CD} + d_{BC} + d_{DA} \\ \mathrm{D}_2 &= d_{AB} + d_{CD} + d_{BD} + d_{AC}. \end{aligned}$$

Thus, we have

$$\begin{aligned} \Delta D &= D_2 - D_1 = (d_{BD} + d_{AC}) - (d_{BC} + d_{DA}) \\ \Delta D &= (d_{OA} + d_{OC} + d_{OB} + d_{OD}) - (d_{BC} + d_{DA}) \\ &= (d_{OA} + d_{OD} - d_{DA}) + (d_{OB} + d_{OC} - d_{BC}) \end{aligned}$$

Moreover, as we have known, the sum of two edges is bigger than one edge in a triangle; thus,

$$\Delta D > 0$$

namely, $D_1 < D_2$.

Therefore, a convex hull in a quadrangle is the shortest total length of links.

Suppose that when $n = N$, the convex hull $D_{Shell(N)}$ is the shortest link.

Then, as shown in Figure 7(c), an N-polygon can be divided into several triangles and quadrangles, and when $n = N + 1$, $D_{N+1} = D_{Shell(N)} + D_{\Delta AN(N+1)} - d_{AN}$. The convex hull of a triangle has been verified to be the minimum; thus,

$$\min D_{N+1} = D_{Shell(N+1)}.$$

In the meantime, when $n = N + 2$, $D_{N+2} = D_{Shell(N)} + D_{\Delta AN(N+2)} - d_{AN}$, and the convex hull of a quadrangle has been verified to be the minimum as well; thus,

$$\min D_{N+2} = D_{Shell(N+2)}.$$

Therefore, that convex-hull crystal shell is the optimal way to link straightforwardly. ■

Lemma 2. During precipitation, the energy released by the lake center attracts crystals from near to far from the crystal shell. Thus, the crystal grows from shore to center.

Proof. According to the energy function from the lake center to crystals in Definition 7,

$$P_i(t) = \frac{\alpha \mathbb{P}}{d_{i0}} = \frac{\alpha \mathbb{P}}{||L_i - L_0||}.$$

Thus, we may obtain

$$P_i'(t) = \frac{\partial \mathbb{p}}{\partial d_{i0}} = -\alpha \frac{\mathbb{P}}{d_{i0}^2} = -\alpha \frac{\mathbb{P}}{||L_i - L_0||^2} < 0.$$

$P_i(t)$ decreases with the increase of d_{i0}; thus, the energy that crystals absorbed from the lake is inversely proportional to its distance from the center. Therefore, a crystal farther from the center, whose energy is lower, is easier to precipitate. The crystal precipitates from shore to center and converges to the lake center by the energy function. ■

Lemma 3. During the process of growth, the crystal links to the crystal shell with the minimal length of link formed by the energy function.

Proof. According to the energy function from the frozen crystal to the unfrozen crystal,

$$S_i(t) = \max[S_{ji}(t)].$$

Suppose that $d_S = ||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||$.
Taking the logarithm for both sides leads to

$$\frac{\partial S_{ji}(t)}{\partial D_S} = -\beta \frac{\mathbb{S}}{{d_S}^2} < 0.$$

It indicates that the energy absorbed from the frozen crystal $S_{ji}(t)$ is inversely proportional to d_S and $S_i(t) = \max[S_{ji}(t)]$.

Therefore, as long as we can verify that equation $\min\{||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||\}$ can attract the crystal into the crystal shell in the optimal way, we may verify the conclusion.

Set the growth process in Figure 8 as an example; crystal c_A precipitates and joins into the crystal convex shell $BCDE$ in two bonds to form a new crystal shell.

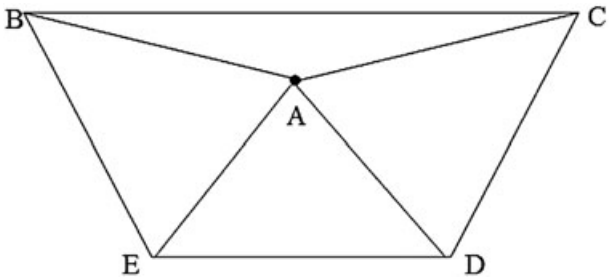


FIGURE 8. The crystal growth.

Assume that the total length of original crystal is

$$D = d_{BC} + d_{CD} + d_{DE} + d_{EB}.$$

If crystal c_A joins the crystal link BC, the total length of crystal links D' is

$$\begin{aligned} D' &= d_{AB} + d_{AC} + d_{CD} + d_{DE} + d_{BE} \\ &= D + d_{AB} + d_{AC} - d_{BC}. \end{aligned}$$

If crystal c_A joins the crystal link DE, the total length of crystal links D'' is

$$\begin{aligned} D'' &= d_{AE} + d_{AD} + d_{DE} + d_{BE} + d_{CD} \\ &= D + d_{AE} + d_{AD} - d_{DE}. \end{aligned}$$

The minimum is $\mathbb{D} = \min\{D', D''\}$. Assume that two neighbor crystals are L_j and L_{j+1}, respectively; thus, $\mathbb{D} = \min\{D', D''\} = \min\{||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||\}$, which is matched with the energy function exactly.

Therefore, we can prove that the crystal joins the crystal shell with the minimal links by the energy function during the process of growth. ■

Theorem 1. The crystal link function is convergent in the CEO algorithm.

Proof. The main stages of freezing are formation of the crystal shell, precipitation, and growth of the crystal.

(1) According to Lemma 1, the connection way of the convex hull has the shortest link length.
(2) Lemma 2 illustrates that the energy released by the lake center attracts crystals that are close to or distant from the crystal shell, and the crystal grows from the shore of the lake to the center during precipitation.
(3) Finally, Lemma 3 points out that the crystal joins the crystal shell with the minimal link by the energy function during the process of growth.

Therefore, when the lake is totally frozen, we have proved that the link length is shortest. Thus, the crystal links satisfy the conditions of convergence. The conclusion thus is straightforward. ■

Lemma 4. The search scale of the CEO algorithm is global.

Proof. According to Definition 1, if $Crystal - \varepsilon = \{c_1, \ldots, c_i\}$, we have

$$Crystal - \varepsilon = CrystalW - \varepsilon + CrystalI - \varepsilon.$$

When $t = 0$, $CrystalI - \varepsilon = \phi$ and $CrystalW - \varepsilon = Crystal - \varepsilon$. And when $t = end$, $CrystalI - \varepsilon = Crystal - \varepsilon$ and $CrystalW - \varepsilon = \phi$.

After an iteration of the CEO algorithm, all the crystals are traversed. Thus, the global search scale is verified. ■

Lyapunov Stability Theorem. The Second Stability Theorem, which is almost universally used nowadays, makes use of a Lyapunov function $V(x)$, which has an analogy to the potential function of classical dynamics. It is introduced as follows for a system having a point of equilibrium at $x = 0$. Consider a function $V(x) : R^n \to R$ such that $V(x) \geq 0$ with equality if and only if $x = 0$ (positive definite) and $V(x) = \frac{d}{dx}V(x) \leq 0$ with equality if and only if $x = 0$ (negative definite).

Then $V(x)$ is called a Lyapunov function candidate, and the system is asymptotically stable in the sense of Lyapunov.

Theorem 2. The total energy function is bounded and decreases monotonically in the CEO algorithm.

Proof. Prove the convergence of crystal function by the Lyapunov stability theorem. Construct the Lyapunov function:

$$V(p,t) = \sum_{i=0}^{n} E_i(t) = \sum_{i=0}^{n} (E_{i0} + Q_i(t)).$$

The crystal releases energy when freezing; thus, $\sum_{i=0}^{n} (Q_i(t)) < 0$.

$$|V(p,t)| = |\sum_{i=0}^{n} (E_{i0} + Q_i(t))| \leq \sum_{i=0}^{n} |(E_{i0})|.$$

Therefore, $V(p,t)$ is bounded.
Furthermore,

$$V(p,t) = \sum_{i=0}^{n} (E_{i0} + \lambda_1 P_i(t) + \lambda_2 S_i(t) + \lambda_3 W_i(t)).$$

Substitute the energy function:

$$V(p,t) = \sum_{i=0}^{n} \left(E_{i0} + \lambda_1 \frac{\alpha \mathbb{P}}{||L_i - L_0||} + \lambda_2 \frac{\beta \mathbb{S}}{||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||} + \lambda_3 \gamma \mathbb{C} \right)$$

$$\frac{\partial V(p,t)}{\partial t} = \sum_{i=0}^{n} \left(\lambda_1 \frac{\alpha}{||L_i - L_0||} \cdot \frac{\partial \mathbb{P}}{\partial t} + \lambda_2 \frac{\beta}{||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||} \cdot \frac{\partial \mathbb{S}}{\partial t} + \lambda_3 \gamma \cdot \frac{\partial \mathbb{C}}{\partial t} \right).$$

According to the flow of the CEO model (Figure 2) and Definition 5, after the initialization of crystals, we obtain $\lambda_1 << \lambda_2, \lambda_1 << \lambda_3$ in the process of precipitation. Thus, the center heat is ignored during the iteration, so let $\lambda_1 = 0$ to simplify the equation, by

$$\frac{\partial V(p,t)}{\partial t} = \sum_{i=0}^{n} \left(\lambda_2 \frac{\beta}{||L_j - L_i|| + ||L_{(j+1)} - L_i|| - ||L_j - L_{(j+1)}||} \cdot \frac{\partial \mathbb{S}}{\partial t} + \lambda_3 \gamma \cdot \frac{\partial \mathbb{C}}{\partial t} \right).$$

And $\mathbb{S} < 0$ and $\mathbb{C} < 0$. Moreover, all the other parameters is above 0. Thus, we obtain

$$\frac{\partial V(p,t)}{\partial t} \leq 0$$

which means that $V(p,t)$ decreases monotonically by time.

Therefore, that the energy function is bounded and decreases monotonically in the CEO algorithm has been verified. ■

Convergence Theorem (Kellogg et al. 1929) Supposing that A is an algorithm on X and Ω is the set of solution, the initial point is given as $x^{(1)} \in X$, and iterations are as follows.

If $x^{(k)} \in \Omega$, the iteration is finished; or set $x^{(k+1)} \in A(x^{(k)})$.

Let $k + 1$ substitute k, and repeat the preceding process. Thus, we obtain the sequence $\{x(k)\}$.

Then set the following:

(1) Sequence $\{x(k)\}$ is contained in compact subset X.
(2) There exists a continuous function, which is a decreasing function of Ω and A.
(3) Mapping A is closed on the complement of Ω.

The limitation of any convergent subsequence of sequence $\{x(k)\}$ belongs to Ω.

Theorem 3. The total energy function $E(t)$ is convergent and can reach the stable point.

Proof. According to the convergence theorem, we verify the convergence of the CEO algorithm from the following aspect:

(1) It has been verified that the search scale of the CEO algorithm is global, which confirms that the crystal set $\{Crystal - \varepsilon_i\}$ obtained in each iteration is contained in a compact subset $Crystal - \varepsilon$.
(2) The fact that there exists a continuous ordered function $E(t)$ has been proved in Theorem 4. Moreover, the function is a descent function about the crystal solution $\{Crystal - \varepsilon_i\}$ and the initial crystal set $\{c_1, \ldots, c_i\}$.
(3) Set $Crystal - \varepsilon_{optimal}$ as the optimal solution, which is the solution set when all the crystals have precipitated. In addition, the total energy is 0. We obtain a solution set in each iteration of the process of freezing. Before the final optimal solution, the relationship of function mapping is one to one. Therefore, we may conclude that the complement set of the optimal solution is closed.

According to proofs (1–3), we may infer that the energy function of the CEO algorithm satisfies the three rules of the convergence theorem. Therefore, the limitation of convergent subsequence of sequence $\{Crystal - \varepsilon_i\}$ belongs to $\{Crystal - \varepsilon_{optimal}\}$. The conclusion thus is straightforward. ■

5.2. The Effectiveness for TSP Analysis

Theorem 4. The CEO algorithm is suitable for the TSP.

Proof. An optimal traversal order $S = \{s_1, s_2, \ldots, s_n\}$ for cities City $= \{city_1, city_2, ..., city_n\}$ with the minimal length of all the paths passed by is required in the TSP.

Similarly, crystals precipitate in some order and join together by two-bond links in the CEO algorithm. From Theorem 1, we know that the total length of crystal links is convergent and the crystals join together with the shortest links. Moreover, Theorem 3 tells us that the CEO algorithm is a global search algorithm and its energy function can reach stability.

Therefore, because the connection between the CEO algorithm and TSP has been analyzed, it is reasonable to solve TSP with the CEO algorithm. Let the crystal represent the city in TSP. Moreover, the order of crystal precipitation is the order in which we traverse the cities. The total length of crystal links is the length of paths as well. The convergence of crystal links has been verified earlier, so we conclude that the CEO algorithm can find the optimal TSP path.

In summary, the CEO algorithm can solve the TSP. ■

6. SIMULATIONS AND COMPARISONS

In this section, the performance of CEO for TSP is examined and analyzed by simulation and comparison. To begin with, we discuss the selection of key parameters by experimental results. Second, the effectiveness and efficiency of CEO are tested by simulating instances in TSPLIB95, which is a benchmark of TSP. Third, the simulation results are compared with the other effective algorithms for TSP. At last, the parallelism of the CEO is tested by simulations with different numbers of processors.

The simulation environment is in Matlab R2011a on the HPCC of Lenovo Shenteng 6800, whose cluster has eight computation nodes and a console node. Each computation node is a high-performance server whose memory is 24 GB with two 2.4-GHz quad-core CPUs. All the servers' operating system is Red Hat Enterprise Linux 7.

6.1. How to Use the CEO for TSP

First, an example of the overall process and performance in each step is given to explain the solving process of CEO for TSP in detail.

Example 2. Solution of CEO for Ch150 instance in TSPLIB95.

Setting the solution for the Ch150 instance as an example, the solving processes of CEO for TSP are shown in Figure 9. It should be noted that the crystals in the CEO correspond to the cities in TSP. Likewise, crystal links correspond to paths. There are 150 cities in instance Ch150, so 150 crystals are initialized. At the beginning of CEO, the crystal shell is formed after initialization of crystals, as shown in Figure 9(a). From Figure 9(b–e), crystals precipitate successively and join the crystal shell through crystal links. By this means, cities are visited, and paths are selected. After forming the initial shape, a wind-blow iteration takes place. The number of iteration of a wind-blow process T is set as 500 runs. The performance with different values of K is presented in Figure 9(f–h), where $K = 90$ matches the optimal known solution for Ch150 (Figure 9h).

The values of the total length of links of crystals during the iteration, which are also the total length of paths, are shown in Figure 10. Analyzing the performance of the CEO (as shown in Figures 9 and 10), we may draw the following inferences.

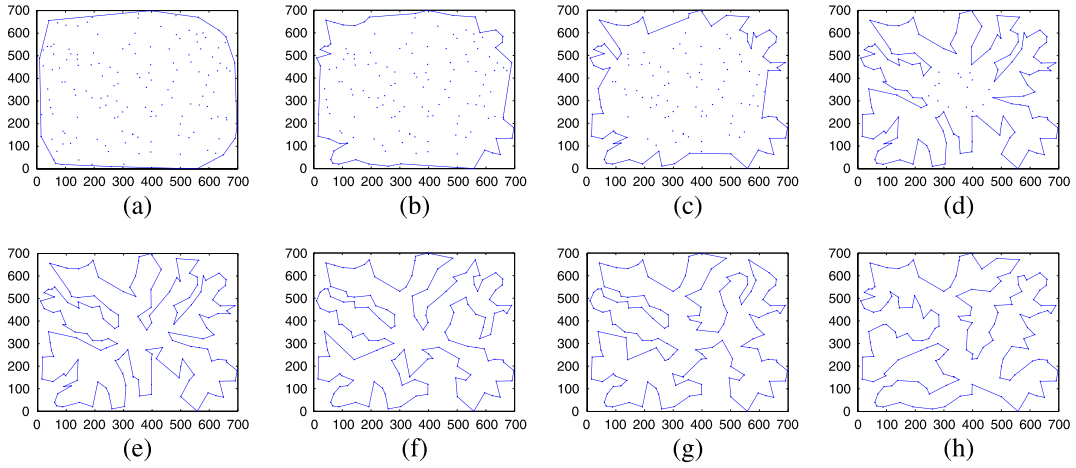


FIGURE 9. Solving process and performance of the crystal energy optimizer for the traveling salesman problem with different values of K.

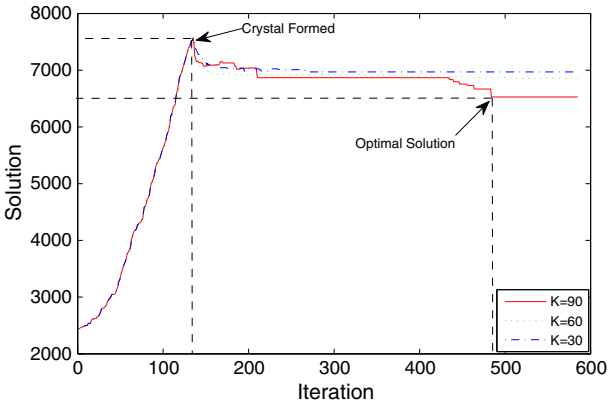


FIGURE 10. Solution process of crystal energy optimizer on a280 with different values of K.

(1) The process of growth of crystals is from outside to inside, including all the crystals on the lake.
(2) A suboptimal solution is obtained by the initial shape of the crystal, which is unrelated with K.
(3) Wind-blow can optimize the solution further.
(4) The performance of the final result is related with K and T.
(5) The running time is contributed by both processes of formation and wind-blow. Therefore, running time will grow with the increase of K and T. Moreover, if $K \gg N$ (number of cities), the running time is mainly determined by K and T.

6.2. Parameters Analysis

The parameters of the CEO can be classified into two categories according to the two main processes of algorithm, which are precipitation–growth and wind-blow.

For the precipitation–growth process, scale factors α, β, and γ in equations (10), (12), and (13) are uniform random numbers in the range of (0, 1). Decision factors λ_1, λ_2, and λ_3 in equation (7) determine the dominant role of energy factors in different processes. Therefore, their values vary in different steps. In detail, decision factor λ_1, which decides the role of center energy, will be set as $\lambda_1 = 100$ in the process of precipitation and $\lambda_1 = 100$ in the process of growth. On the contrary, decision factor λ_2, which maintains the influence of the energy of frozen crystals, will be set as $\lambda_2 = 0$ when crystals precipitate and $\lambda_2 = -100$ when they grow. In addition, decision factor λ_3, which represents the weight of the wind, is constant as $\lambda_3 = -50$ for all processes. These parameters are key factors that make sure crystals precipitate and grow in a proper way. The summary of parameters is shown in Table 3.

On the other hand, for the wind-blow process, two key parameters, number of melting crystals for each iteration K and total iteration number T, have a pivotal influence on the overall performance of CEO. Besides, the setting of total iteration number T should take the total number of cities into consideration. It is set as $T = 500$ when the number of cities is less than 2,000 and $T = 1{,}000$ when the number of cities is larger than 2,000. A number of different alternative values are tested. Those that give the best computational results concerning both the quality of the solution and the computational time are selected.

Thus, we mainly discuss the impact and selection of K in this subsection. To test the effectiveness of different K, the performance of CEO is tested on 50 TSP benchmark instances.

TABLE 3. Parameter Settings.

Parameters	Values	Meanings
α, β, γ	[0, 1]	Scale factors
λ_1	100 (in the process of precipitation); 0 (in the process of growth)	Decision factor for center energy
λ_2	0 (in the process of precipitation); −100 (in the process of growth)	Decision factor for frozen crystal
λ_3	−50	Decision factor for wind
K	55% * *CityNum*	Number of melted crystal for each iteration
T	500 (for small and medium scales); 1,000 (for large scale)	Iteration number of wind-blow

TABLE 4. Solution and Running Time of Crystal Energy Optimizer on Berlin52 with Different Values of K.

	Solution			CPU (s)	
K	AVG	BEST	Std.	AVG	BEST
0	8,315.860	8,315.86	0	0.07	0.06
1	8,253.392	8,241.82	12.21	0.09	0.06
5	8,195.646	7,850.44	99.13	0.07	0.05
10	8,098.300	7,607.93	151.53	0.06	0.05
15	8,064.664	7,688.79	149.66	0.07	0.07
20	8,070.367	7,753.16	148.86	0.17	0.09
25	8,031.522	7,678.26	173.20	0.32	0.24
30	8,010.475	**7,542.21**	179.24	0.37	0.30
35	**7,985.700**	7,542.53	159.22	0.37	0.28
40	8,026.739	7,618.08	191.59	0.53	0.33
45	8,010.969	7,675.20	178.34	0.53	0.33
50	7,994.735	7,673.22	161.86	0.42	0.37

Three examples (Berlin52, Ch150, and a280) are shown in Tables 4–6 (the optimal solution has been highlighted). For running time, it is obvious that both average and best running time will increase with the increase of K. However, the change of the solution with K is not monotonous. The best solution is not obtained by maximal K. For example, $K = 30$–35 obtains the best solution for Berlin52, $K = 80$–90 for Ch150, and $K = 150$ for a280. It should be noted that a higher K also brings a higher standard deviation.

Because different optimal K is gained in different TSP instances, to find the relationship between optimal K and number of cities, two factors (quality and percent number) are introduced. Thus, the solutions of different instances and its corresponding K are uniform.

$$\omega = (Solution - OPT) * 100/OPT \tag{18}$$

$$PercentNum = K/TotalNum \tag{19}$$

where ω give the quality of the solution, OPT is the best known solution, and the total number of cities is given by $TotalNum$. Therefore, Figure 11 shows the performance (quality

TABLE 5. Solution and Running Time of Crystal Energy Optimizer on Ch150 with Different Values of K.

	Solution			CPU (s)	
K	AVG	BEST	Std.	AVG	BEST
0	7,531.920	7,531.92	0	0.55	0.41
1	7,312.917	7,195.94	56.38	0.59	0.57
5	7,099.275	7,016.43	27.12	0.62	0.44
10	7,071.178	6,882.09	48.75	0.70	0.63
20	7,041.349	6,870.89	72.55	0.85	0.71
30	7,034.675	6,775.12	84.63	0.98	0.69
40	7,010.620	6,809.84	81.35	1.24	0.95
50	7,010.253	6,735.42	91.37	1.42	0.89
60	6,987.152	6,794.28	81.52	1.72	1.31
70	6,988.371	6,684.98	92.53	1.84	1.13
80	6,971.472	**6,528.10**	95.34	2.26	1.55
90	**6,951.810**	6,528.30	97.86	1.61	0.89
100	6,961.911	6,786.99	82.88	1.50	1.01
110	6,973.996	6,760.25	94.41	3.26	2.50
120	6,962.659	6,760.71	104.01	3.49	2.30

TABLE 6. Solution and Running Time of CEO on a280 of TSPLIB with Different Values of K.

	Solution			CPU (s)	
K	AVG	BEST	Std.	AVG	BEST
0	3,081.000	3,081.00	0	0.98	0.87
30	2,736.453	2,699.94	16.16	1.33	0.86
60	2,731.700	2,686.25	20.65	1.68	1.19
90	2,729.252	2,672.18	19.70	3.14	1.61
120	2,728.575	2,683.27	15.37	4.09	2.01
150	**2,719.460**	**2,579.21**	23.07	6.49	2.78
180	2,728.987	2,579.90	32.57	9.07	4.08
210	2,739.320	2,633.12	42.13	9.91	5.13
240	2,742.510	2,681.39	53.78	10.32	5.88
270	2,761.440	2,691.99	66.91	12.18	8.21

of the solution) of CEO with different percent values of K on different TSP instances. The best range of K is 55–65% of the total number of cities.

6.3. Effectiveness of the CEO

The algorithm is tested on a set of 45 Euclidean sample problems with sizes ranging from 100 to 7,397 nodes. Each instance is described by its TSPLIB name and size. For example, in Table 7, the instance named Rd400 has a size equal to 400 cities. Data in the crystal formed columns are obtained before wind-blow, and data in wind-blow columns are

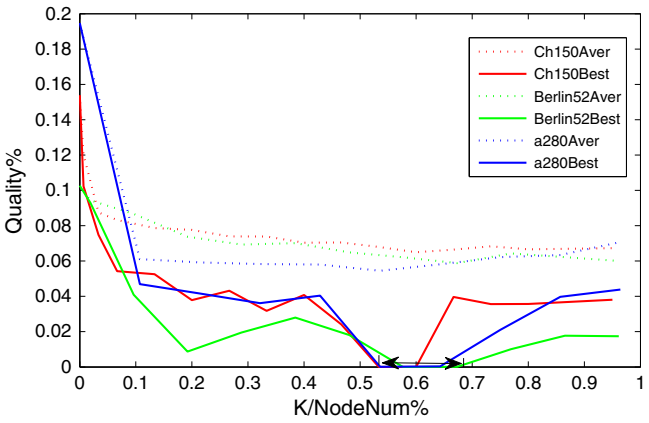


FIGURE 11. Performance of the algorithm with different values of K on different traveling salesman problem instances.

TABLE 7. Performance of Crystal Energy Optimizer on TSPLIB95.

		Crystal formed			Wind-blow				
Instance	OPT	Solution	ω (%)	CPU (s)	Best	AVG	ω_{AVG} (%)	ω_{BST} (%)	CPU (s)
Rd400	15,281	16,108	5.1341	3.06	15,281	15,281	0	0	13.16
Fl417	11,861	12,654	6.2668	1.35	11,861	11,922	0.5143	0	9.82
Pr439	107,217	110,900	3.321	2.67	107,217	107,217	0	0	8.73
Pcb442	50,778	60,019	15.397	4.59	50,778	50,778	0	0	15.23
D493	35,002	41,853	16.369	5.21	35,002	35,002	0	0	16.52
Rat575	6,773	7,203	5.9697	7.8	6,773	6,782	0.1329	0	18.07
P654	34,643	37,758	8.2499	8.71	34,643	34,643	0	0	19.89
D657	48,912	49,786	1.7555	3.41	48,912	48,912	0	0	19.66
Rat783	8,806	9,113	3.3688	6.06	8,806	8,825	0.2158	0	27.5
dsj1000	18,659,688	18,676,291	0.0889	8.45	18,659,688	18,659,979	0.0016	0	37.69
Pr1002	259,045	261,211	0.8292	7.91	259,045	259,045	0	0	40.32
u1060	224,094	227,928	1.6821	11.3	224,094	224,094	0	0	47.8
vm1084	239,297	239,822	0.2189	10.9	239,297	239,297	0	0	59.3
Pcb1173	56,892	57,001	0.1912	9.3	56,892	56,892	0	0	60.9
D1291	50,801	53,113	4.353	11.7	50,801	50,825	0.0472	0	63.1
Rl1304	252,948	255,213	0.8875	12.8	252,948	252,948	0	0	80.3
Rl1323	270,199	278,211	2.8798	12.6	270,199	270,199	0	0	84.1
nrw1379	56,638	57,221	1.0189	20.6	56,638	56,640	0.0035	0	89.6
Fl1400	20,127	22,121	9.0141	18.9	20,127	20,127	0	0	83.2
u1432	152,270	156,311	2.5852	26.2	152,270	152,270	0	0	98
Fl1577	22,249	23,813	6.5678	27.9	22,249	22,258	0.0405	0	87.4
d1655	62,128	62,975	1.345	32.9	62,130	62,141	0.0209	0.00322	94.5
vm1748	336,556	339,200	0.7795	33.1	336,556	336,571	0.0045	0	108.7
u1817	57,201	59,311	3.5575	30.2	57,201	57,201	0	0	103
Rl1889	316,536	318,215	0.5276	37.8	316,536	316,564	0.0088	0	114.4
D2103	80,450	86,991	7.5192	40.4	80,450	80,484	0.0423	0	109.1
u2152	64,253	65,116	1.3253	44.8	64,253	64,318.4	0.1018	0	110.9
u2319	234,256	234,513	0.1096	50.3	234,256	234,298	0.0179	0	136.9
Pr2392	378,032	378,387	0.0938	61.1	378,032	378,122	0.0238	0	164.7
pcb3038	137,694	155,275	11.322	84.9	137,922	138,620	0.6725	0.16558	212.5
fl3795	28,772	30,640	6.0966	98.9	28,801	28,898	0.4379	0.10079	245.2
fnl4461	182,566	183,612	0.5697	102.2	182,998	187,641	2.7798	0.23663	311.6
rl5915	565,530	596,211	5.146	131.8	577,112	579,321	2.4386	2.04799	381.7
rl5934	556,045	578,006	3.7994	342.9	560,144	570,001	2.5099	0.73717	431.3
pla7397	23,260,728	23,921,000	2.7602	622.7	23,261,231	23,591,110	1.4203	0.00216	629.3
AVG	1,338,407.66	1,361,802.91	4.03	55.3	1,338,889.8	1,348,835.04	0.33	0.09	118.12

the final solution. All the experiments in this subsection are repeated 500 times. Table 7 shows the experimental results for sizes of instances lager than 400. ω_{BST} is the best quality of the solution and ω_{AVG} is the average quality of the solutions. By analyzing Table 7, the following inferences and conclusions are drawn.

(1) The best solution obtained in all but seven instances equals the optimal solution, making an 80% successful rate for medium and large data. For the other seven instances, only the quality of the solution of rl5915 is 2%, while quality of all the others is less than 1%.
(2) The average solution obtained in 17 instances equals the optimal solution. For the other instances, only the CEO has not found the optimum in one or two runs. However, even if the best solution is not found in all runs, the solutions obtained by CEO are very close to the best solutions.
(3) Judging from the CPU time needed, which is relatively low, the CEO is very efficient.
(4) Another important character of the CEO is that the suboptimal solution obtained by the crystal formation process is relatively acceptable and with extremely high efficiency. The average solution quality in this process is 4.03%, which is an acceptable solution in some cases and even better than some well-known algorithms. (Compared data are shown in the next subsection.) However, the time consumed is only 20–50% of the final solution. This high efficiency is of great significance in many applications in real life, such as GPS in mobile vehicles, which has a high requirement for speed of solution but relatively less for accuracy.

    Furthermore, it should also be noticed that to present a very efficient and effective algorithm, the selection of parameters, particularly K and T in this case, takes both running time and convergence into consideration. If we increase the number of melting crystals K and the number of iteration T, better solutions with higher time consumption could be acquired.

6.4. Comparisons between CEO and Other State-of-the-Art Algorithms

To evaluate the effectiveness and efficiency of CEO, the performance and solution are analyzed furthermore in this subsection. The experiments are divided into three groups as small-scale, medium-scale, and large-scale groups according to the size of cities in TSPLIB. Comparisons with other algorithms are performed. However, limited by the data given by references, we have to compare with different algorithms in different scale groups. The referred algorithms used in this subsection are state-of-the-art algorithms from references in recent years.

6.4.1. Small Scale: 100–500. For the small-scale instances, 500 different runs with the selected parameters (in Table 8) were performed for each of the benchmark instances. The solutions for the same instances of CEO and three other algorithms, honey-bee mating optimization (HBMO) (Marinakis, Marinaki, and Dounias 2011), greedy subtour mutation (GSTM) (Albayrak and Allahverdi 2011), and populated iterated greedy algorithm with inver-over operator (IGP_IO) (Tasgetiren et al. 2013), are shown in Table 8. From Table 8 (the optimal solution has been highlighted), the following could be observed:

(1) CEO and HBMO can always find the optimal solution in a short time for all the instances, while GSTM and IGP_IO cannot.
(2) The HBMO is a competitive algorithm with all the best solution equaling optimum; however, the running time of CEO is almost half of that of HBMO. Figure 12 shows the average time consumed by each algorithm. The running time of IGP_IO and CEO is far lower than that of the other two. In addition, CEO's is a little lower than IGP_IO's as well.

    Therefore, considering both the quality and running time of the solution, CEO has the best performance of all in the small-scale group.

TABLE 8. Comparison of Performance between CEO and HBMO, GSTM, and IGP_IO.

Instance	HBMO (Marinakis et al. 2011)			GSTM (Albayrak and Allahverdi 2011)			IGP_IO (Tasgetiren et al. 2013)			CEO		
	ω_{AVG} (%)	ω_{BST} (%)	CPU (s)	ω_{AVG} (%)	ω_{BST} (%)	CPU (s)	ω_{AVG} (%)	ω_{BST} (%)	CPU (s)	ω_{AVG} (%)	ω_{BST} (%)	CPU (s)
KroA100	0	0	0.62	1.1836	0	6.987	0	0	0.47	**0**	**0**	**0.17**
Pr144	0	0	1.68	1.809	0	13.598	0.35	0	0.43	**0**	**0**	**0.21**
Ch150	0	0	2.01	0.6357	0.4596	11.24	0.3	0.25	0.87	**0**	**0**	**1.55**
Pr152	0	0	2.21	1.6202	0.7695	7.937	0.51	0	2.09	**0**	**0**	**1.63**
Rat195	0	0	3.45	1.8425	0.6027	15.05	1.48	1.08	2.83	**0**	**0**	**2.58**
D198	0	0	4.27	1.2193	0.3866	12.096	0.65	0.43	3.5	**0**	**0**	**2.65**
KroA200	0	0	5.01	1.5432	0.8683	13.292	0.52	0.34	3.77	**0**	**0**	**2.59**
Ts225	0	0	5.38	0.4994	0.2527	11.559	0.2	0	3.14	**0**	**0**	**2.6**
Pr226	0	0	5.41	1.5287	0.7242	13.843	0.33	0	3.53	**0**	**0**	**2.67**
Pr299	0	0	8.67	2.9169	1.2326	17.424	1.04	0.28	6.21	**0**	**0**	**2.98**
Pcb442	0	0	37.12	2.7758	2.0501	19.132	1.42	0.98	15.33	**0**	**0**	**15.23**
AVE	0	0	6.8936	1.5977	0.6678	12.9235	0.6182	0.3055	3.833	**0**	**0**	**3.169**

CEO, crystal energy optimizer; GSTM, greedy subtour mutation; HBMO, honey-bee mating optimization; IGP_IO, populated iterated greedy algorithm with inver-over operator.

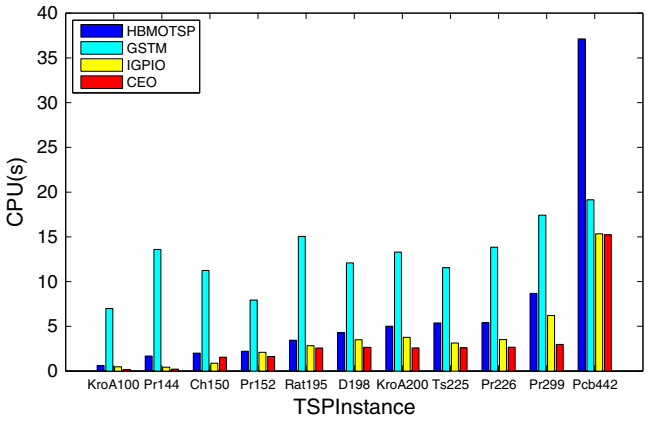


FIGURE 12. Comparison of running time.

TABLE 9. Comparison of Quality of Solutions for Traveling Salesman Problem between CEO and HK-ACO, MMAS, and HBMO (Standard Quality for Ranking, Average Results from 1,000 runs).

	HK-ACO (Li, Ju, and Zhang 2008)		MMAS (Li et al. 2008)		HBMO (Marinakis et al. 2011)		CEO	
	Best	AVG	Best	AVG	Best	AVG	Best	AVG
Pr1002	0.05057	0.085545	0.044008	0.378313	0	0.001	**0**	**0**
Pcb1173	0.008789	0.137981	0.021093	0.100893	0	0.003	**0**	**0**
D1291	0	0.014567	0	0.091927	0	0	**0**	**0**
Rl1304	0	0.083416	0	0.229363	0	0	**0**	**0**
Rl1323	0.009993	0.136566	0.020355	0.194856	0	0	**0**	**0**
Fl1400	0.168927	0.504049	0.462066	0.855418	0	0.011	**0**	**0**
Fl1577	0.395523	0.736348	0.053935	0.785069	0	**0.022**	**0**	0.041
Rl1889	N/A	N/A	N/A	N/A	0	0.017	**0**	**0.009**
D2103	N/A	N/A	N/A	N/A	0	**0.041**	**0**	0.043
Pr2392	N/A	N/A	N/A	N/A	0	0.026	**0**	**0.018**
AVE	0.090543	0.242639	0.085922	0.376549	0	0.0121	**0**	**0.0111**

CEO, crystal energy optimizer; HBMO, honey-bee mating optimization; HK-ACO, Held–Karp ant colony optimizer; MMAS, MAX–MIN ant system; N/A, the result is not mentioned in the reference. The optimal solution for each instance has been highlighted in the table.

6.4.2. Medium Scale: 500–2,000. The medium-scale experiments choose the instances with sizes from 500 to 2,000 from TSPLIB. Comparisons among the best-known improved ACO, such as Held–Karp ACO and MAX–MIN ant system, and the HBMO are performed in Table 9.

Marinakis, Migdalas, and Pardalos (2005, 2011) raise a ranking among the best-known algorithms from tour constructions heuristics, simple local search algorithms, variable neighborhood search implementations, and meta-heuristics for TSP. The standard for ranking is their average quality over 10 instances with a size over 1,000, which are Pr1002, Pcb1173, D1291, Rl1304, Rl1323, Fl1400, Fl1577, Rl1889, D2103, and Pr2392. Because the computational speed is mainly affected by the compiler and hardware, the comparisons and ranking with the algorithms from the literature are performed only in terms of the quality of the solutions. Therefore, we run CEO over these 10 instances and have the best and average solutions from 500 runs, as shown in Table 9.

The proposed CEO algorithm is ranked in third place among 55 algorithms. (The top 30 are presented in Table 10, and for other details, readers should refer to Marinakis et al. (2011,

TABLE 10. Ranking of Algorithms.

No.	Method	Average
1	Tour merging (Applegate, Cook, and Rohe 2003)	0.0034
2	PHGA (Nguyen et al. 2007)	0.0066
3	**CEO**	**0.0111**
4	HBMO for TSP	0.0121
5	ILK-NYYY (Johnson and McGeoch 2007)	0.0354
6	KHs MTV on LK (Helsgaun 2000)	0.0367
7	ILK-J (Johnson and McGeoch 1997)	0.0853
8	ALK (Johnson and McGeoch 2007)	0.48
9	Concorde CLK (Applegate et al. 2003)	0.4863
10	I-3-Opt-J (Johnson and McGeoch 1997)	0.514
11	ACR chained LK (Applegate et al. 2003)	0.5827
12	ILK-N (Neto 1999)	0.5924
13	Helsgaun LK (Helsgaun 2000)	0.669
14	BSDPH (Helsgaun 2000)	0.71
15	LK-HK-Christo-starts (Helsgaun 2000)	1.051
16	TS-LK-DB (Zachariasen and Dam 1996)	1.06
17	TS-SC-DB (Zachariasen and Dam 1996)	1.111
18	ENS-GRASP (Marinakis et al. 2005)	1.181
19	LK-NYYY (Helsgaun 2000)	1.228
20	Johnson LK (Johnson and McGeoch 1997)	1.388
21	TS-LK-LK (Zachariasen and Dam 1996)	1.475
22	TS-SC-SC (Zachariasen and Dam 1996)	1.483
23	Neto LK (Neto 1999)	1.547
24	SC EC (Johnson and McGeoch 2007)	1.66
25	VNS-3-Hyperopt (Johnson and McGeoch 2007)	2.061
26	Concorde-LK (Applegate et al. 2003)	2.778
27	VNS-2-Hyperopt (Helsgaun 2000)	2.874
28	Applegate LK (Applegate et al. 2003)	3.2
29	3opt-B (Bentley 1992)	3.288
30	3opt-J (Johnson and McGeoch 1997)	3.392

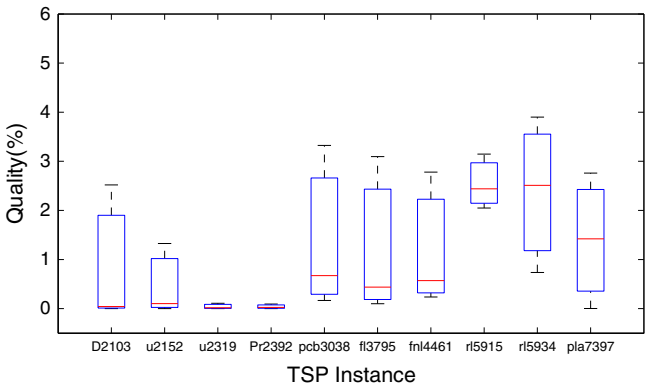


FIGURE 13. Performance of the CEO for over 2,000-city instance in TSPLIB95.

TABLE 11. Results of Parallel Experiments with Different Numbers of Processors.

	1 processor	2 processors		4 processors		8 processors		12 processors		16 processors		20 processors	
City	Time (s)	Time (s)	Ratio	Time (s)	Ratio	Time (s)	Ratio	Time (s)	Ratio	Time (s)	Ratio	Time (s)	Ratio
500	14.72	7.37	1.997	3.71	3.968	1.86	7.914	1.23	11.967	0.93	15.828	0.74	19.891
1,000	33	16.52	1.998	8.27	3.990	4.26	7.746	2.76	11.957	2.09	15.789	1.68	19.642
1,500	52.33	26.31	1.989	13.23	3.955	6.52	8.026	4.38	11.947	3.28	15.954	2.64	19.822
2,000	71.49	35.79	1.997	17.94	3.985	8.94	7.997	5.96	11.995	4.51	15.851	3.74	19.115
2,500	90.9	45.54	1.996	22.75	3.996	11.43	7.953	7.74	11.744	5.79	15.699	4.68	19.423
3,000	110.34	55.28	1.996	27.65	3.991	14.01	7.876	9.43	11.701	7.12	15.497	5.69	19.392
3,500	130.42	65.28	1.998	32.75	3.982	16.53	7.890	11.23	11.614	8.37	15.582	6.73	19.379
4,000	150.76	75.39	2.000	37.79	3.989	19.05	7.914	12.77	11.806	9.73	15.494	7.93	19.011
4,500	170.33	85.36	1.995	42.84	3.976	21.84	7.799	14.56	11.698	11.12	15.317	8.98	18.968
5,000	190.49	95.43	1.996	48.02	3.967	24.42	7.801	16.43	11.594	12.45	15.300	10.12	18.823
6,000	232.54	116.59	1.995	58.36	3.985	29.59	7.859	20.01	11.621	15.07	15.431	12.34	18.844
7,000	272.7	136.98	1.991	69.81	3.906	34.89	7.816	23.57	11.570	17.75	15.363	14.37	18.977
8,000	312.04	157.39	1.983	79.1	3.945	40.02	7.797	26.96	11.574	20.31	15.364	16.33	19.108
9,000	353.17	177.61	1.988	89.6	3.942	45.49	7.764	30.64	11.526	23.35	15.125	18.88	18.706
10,000	394.88	198.2	1.992	100.71	3.921	50.92	7.755	34.42	11.472	25.76	15.329	20.84	18.948

2005.) Table 10 illustrates that our CEO performs better than most of the tour constructions heuristics, simple local search algorithms, variable neighborhood search implementations, and meta-heuristics.

6.4.3. Large Scale: Larger than 2,000. For the large-scale group with size over 2,000, CEO still has a relatively good performance with the average quality of the solution of less than 3%, while most of the other existing algorithms have difficulty in solving large-scale TSP. To illustrate the quality of solutions of CEO, the box figure in Figure 13 is the statistics over 500 runs. The average and best solutions of CEO are already shown in Table 7. The best quality of the solutions (all but two instances of less than 1% and two instances of less than 2%) and the average quality of solutions (less than 3%) verify the effectiveness of CEO, while the short distance of the box (at most 2%) also indicates the robustness of CEO.

6.5. Efficiency and Parallelism of the CEO

In this section, we will test and analyze CEO's parallelism. The parameters are the same as listed in Table 3. The iteration number of wind-blow is set as $T = 1{,}000$. The range of the number of cities is from 500 to 10,000. The positions of cities are obtained randomly. We test the parallel effect of CEO when the number of processors used is 1, 2, 4, 8, 12, 16, and 20. For each experiment, the running time under a certain number of iterations and the speedup ratio calculated from it are recorded.

Results in Table 11 are the average results from 500 runs. Figure 14 is the relationship between running time and scale number of cities with different numbers of processors. By analyzing the parallel results, the following conclusions are drawn.

(1) Running time reduces with the increase of the number of processors allocated continuously, while the CEO's parallel processing speed accelerates with it.
(2) With the scale of cities expanding, CEO is able to maintain a stable and high speedup ratio. When the number of cities equals 10,000, the solution could be obtained in 20 s. As a result, the speed of CEO accelerates further, making it more efficient for large-scale problems.

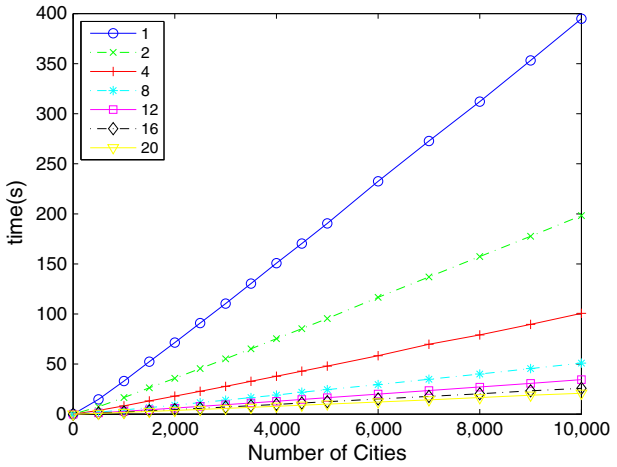


FIGURE 14. Results of parallel experiments.

7. CONCLUSION

A novel nature-inspired parallel algorithm is proposed in this article. The crystal energy algorithm mimics the process of the freezing phenomenon of a lake. Physical and mathematical models are built based on the precipitation rules from publications of *Science* and *Nature*. The CEO algorithm contains two models: precipitation–growth model and wind-blow model. The first model guarantees the global search range and an initial suboptimal solution by crystals precipitating and growing in a proper order. The second model protects the CEO from local optimum effectively. The effectiveness and convergence of our algorithm are verified through the Lyapunov stability theorem and the convergence theorem theoretically. Furthermore, the experimental results of CEO on TSP prove its applicability for solving complex NP-complete problems. Comparing the performance on quality of the solution and computer time of CEO with that of other algorithms, conclusions from the statistical analysis show that the average performance of CEO over all instances is significantly better than that of others, especially for the large-scale instances. Last, a parallelism test for CEO is carried out on a series of TSP instances. Experimental results indicate that CEO has an excellent parallelism property.

In future works, we are going to focus on the improvement and application of our algorithm. First, the wind-blow model is an important part of the CEO, keeping it from local optimum. From our experiments, we also can tell that it is the most time-consuming part as well. Therefore, to improve the performance of the CEO, the wind-blow model could be modified from two aspects. On the one hand, the crystals could be chosen by rules instead of randomly. If we could pick the "bad" crystals wisely, computer time would be saved. On the other hand, the number of crystal chosen K could be set dynamically. Different numbers of crystals dropped in each iteration might improve the effect of the wind. In sum, our major strategy to improve the CEO is to make it more intelligent. Second, CEO is applicable for large-scale complicated problems because of its high parallelism and outstanding quality of solutions. Therefore, we will attempt to solve more real-life complex problems with the CEO in the future.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under grant nos. 60905043, 61073107, and 61173048, the Innovation Program of Shanghai Municipal Education Commission, and the Fundamental Research Funds for the Central Universities.

REFERENCES

ALBAYRAK, M., and N. ALLAHVERDI. 2011. Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Systems with Applications, **38**(3): 1313–1320.

ALTMAN, E., P. NAIN, A. SHWARTZ, and Y. XU. 2013. Predicting the impact of measures against p2p networks: transient behavior and phase transition. IEEE/ACM Transactions on Networking, **21**(3): 935–949.

ANONYMOUS. 2008. Nature's guiding light. Nature Photonics, **2**(11): 639–639.

APPLEGATE, D., W. COOK, and A. ROHE. 2003. Chained Lin–Kernighan for large traveling salesman problems. INFORMS Journal on Computing, **15**(1): 82–92.

BELL, R. E. 2008. The unquiet ice. Scientific American, **298**(2): 60–67.

BENTLEY, J. J. 1992. Fast algorithms for geometric traveling salesman problems. ORSA Journal on Computing, **4**(4): 387–411.

BONABEAU, E., M. DORIGO, and G. THERAULAZ. 2000. Inspiration for optimization from social insect behaviour. Nature, **406**(6791): 39–42.

CUI, Z., D. LIU, J. ZENG, and Z. SHI. 2012. Using splitting artificial plant optimization algorithm to solve toy model of protein folding. Journal of Computational and Theoretical Nanoscience, **9**(12): 2255–2259.

FORREST, S. 1993. Genetic algorithms: Principles of natural selection applied to computation. Science, **261**(5123): 872–878.

HE, S., Q. H. WU, and J. SAUNDERS. 2009. Group search optimizer: An optimization algorithm inspired by animal searching behavior. IEEE Transactions on Evolutionary Computation, **13**(5): 973–990.

HE, S., J. CHEN, P. CHENG, Y. GU, T. HE, and Y. SUN. 2012. Maintaining quality of sensing with actors in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, **23**(9): 1657–1667.

HELSGAUN, K. 2000. An effective implementation of the Lin–Kernighan traveling salesman heuristic. European Journal of Operational Research, **126**(1): 106–130.

ISHTEVA, M., P. A. ABSIL, S. VAN HUFFEL, and L. DE LATHAUWER. 2011. Tucker compression and local optima. Chemometrics and Intelligent Laboratory Systems, **106**(1): 57–64.

JOHNSON, D. S., and L. A. MCGEOCH. 1997. The traveling salesman problem: a case study in local optimization. Local Search in Combinatorial Optimization, **1**: 215–310.

JOHNSON, D. S., and L. A. MCGEOCH. 2007. Experimental analysis of heuristics for the stsp. *In* The Traveling Salesman Problem and Its Variations. Springer: New York, pp. 369–443.

KELLOGG, O. D. ET AL. 1929. Foundations of Potential Theory. Dover: New York.

KEPHART, J. O. 2011. Learning from nature. Science, **2**(5): 682–683.

KIRKPATRICK, S., C. GELATT, and M. VECCHI. 1983. Optimization by simulated annealing. Optimization by Simulated Annealing, **220**(4598): 671–680.

KRINNER, G., J. MANGERUD, M. JAKOBSSON, M. CRUCIFIX, C. RITZ, and J. I. SVENDSEN. 2004. Enhanced ice sheet growth in Eurasia owing to adjacent ice-dammed lakes. Nature, **427**(6973): 429–432.

LI, L., S. JU, and Y. ZHANG. 2008. Improved ant colony optimization for the traveling salesman problem. *In* 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 1, IEEE, Changsha, China, pp. 76–80.

LIM, S., and J. ZHU. 2013. Integrated data envelopment analysis: Global vs. local optimum. European Journal of Operational Research, **229**(1): 276–278.

LIM, W. H., and N. A. MAT ISA. 2014. An adaptive two-layer particle swarm optimization with elitist learning strategy. Information Sciences, **273**: 49–72.

LING, Z., X. FU, W. JIA, W. YU, and D. XUAN. 2011. A novel packet size based covert channel attack against anonymizer. *In* 2011 Proceedings IEEE Infocom, pp. 186–190.

LIU, F., and Z. ZHOU. 2014. An improved QPSO algorithm and its application in the high-dimensional complex problems. Chemometrics and Intelligent Laboratory Systems, **132**(3): 82–90.

MARINAKIS, Y., A. MIGDALAS, and P. M PARDALOS. 2005. Expanding neighborhood grasp for the traveling salesman problem. Computational Optimization and Applications, **32**(3): 231–257.

MARINAKIS, Y., M. MARINAKI, and G. DOUNIAS. 2011. Honey bees mating optimization algorithm for the euclidean traveling salesman problem. Information Sciences, **181**(20): 4684–4698.

MAYBANK, J., and N. N. BARTHAKUR. 1967. Growth and destruction of ice filaments in an electric field. Computational Optimization and Applications, **216**(10): 50–52.

MCKAY, C. P., G. . D CLOW, R. A. WHARTON, and S. W. SQUYRES. 1985. Thickness of ice on perennially frozen lakes. Nature, **313**(2): 561–562.

MEHRJOO, S., A. SARRAFZADEH, and M. MEHRJOO. 2015. Swarm intelligent compressive routing in wireless sensor networks. Computational Intelligence, **31**(3): 513–531.

MURRAY, B. J, D. A. KNOPF, and A. K. BERTRAM. 2005. The formation of cubic ice under conditions relevant to Earth's atmosphere. Nature, **434**(7030): 202–205.

NETO, D. M. 1999. *Efficient cluster compensation for Lin–Kernighan heuristics*, Ph.D. Thesis.

NEUMANN, J. V. 1955. Mathematical Foundations of Quantum Mechanics. No. 2. Princeton University Press: Princeton, NJ.

NGUYEN, H. D., I. YOSHIHARA, K. YAMAMORI, and M. YASUNAGA. 2007. Implementation of an effective hybrid GA for large-scale traveling salesman problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **37**(1): 92–99.

RBOUH, I., and A. E IMRANI. 2014. Hurricane-based optimization algorithm. AASRI Procedia, **6**: 26–33.

SADI, Y., and S. C. ERGEN. 2013. Optimal power control, rate adaptation, and scheduling for UWB-based intravehicular wireless sensor networks. IEEE Transactions on Vehicular Technology, **62**(1): 219–234.

SCHMITT, M., and R. WANKA. 2015. Particle swarm optimization almost surely finds local optima. Theoretical Computer Science, **561**: 57–72.

STRUNK, T. H., C. R. TRACY, and M. KLEIBER. 1973. Perspectives on linear heat transfer. Science, **181**(4095): 184–186.

SUN, J., J. M. GARIBALDI, N. KRASNOGOR, and Q. ZHANG. 2013. An intelligent multi-restart mimetic algorithm for box constrained global optimisation. Evolutionary Computation, **21**(1): 107–147.

TANG, Y., P. JU, H. HE, C. QIN, and F. WU. 2013. Optimized control of DFIG-based wind generation using sensitivity analysis and particle swarm optimization. IEEE Transactions on Smart Grid, **4**(1): 509–520.

TASGETIREN, M. F., O. BUYUKDAGLI, D. KIZILAY, and K. KARABULUT. 2013. A populated iterated greedy algorithm with inver-over operator for traveling salesman problem. *In* Swarm, Evolutionary, and Memetic Computing. Springer: Cham, Switzerland, pp. 1–12.

TEIMOURZADEH, S., and K. ZARE. 2014. Application of binary group search optimization to distribution network reconfiguration. International Journal of Electrical Power & Energy Systems, **62**(11): 461–468.

ZACHARIASEN, M., and M. DAM. 1996. Tabu search on the geometric traveling salesman problem. *In* Metaheuristics. Springer: New York, pp. 571–587.

ZHAN, Z., J. ZHANG, Y. LI, and Y. SHI. 2011. Orthogonal learning particle swarm optimization. IEEE Transactions on Evolutionary Computation, **15**(6): 832–847.

ZHANG, H., Y. ZHU, and H. CHEN. 2014. Root growth model: a novel approach to numerical function optimization and simulation of plant root system. Soft Computing, **18**(9): 521–537.

APPENDIX A: THE CEO FOR CONTINUOUS OPTIMIZATION

The general CEO has been introduced in this article. Its performance on the discrete problem, TSP, has been tested. We attempt to solve continuous optimization problems by CEO in this Appendix.

A1. Description of CEO for Continuous Optimization

Different from the strategy for the discrete problems, mathematical models of CEO have to make some adjustment for the continuous problems. The flowchart of CEO is the same. However, the output at the end of freezing is different. Instead of the links with the least energy between crystals, the center of crystal shell is required to be located at the end of freezing. $f(x)$ is the standard energy function to determine the position of a crystal in the lake. The crystal c_i obtaining min $f(c_i)$ is the center of the crystal shell.

Elaborate descriptions are as follows.

Step 1 (Initialization):

S crystals with n dimensions as $c_i = (c_{i1}, c_{i2}, \ldots, c_{in}) \in \mathbb{R}^n$ and their initial energy E_{i0} are initialized.

Step 2 (Formation of crystal shell):

Tested by the energy function $f(x)$, the last N_S crystals, which are in the lowest positions will precipitate. They will link together to form the crystal shell. However, the frozen crystals will not continue to participate in in the freezing process. To maintain the constant number of crystals, another N_S crystals will be re-initialized in the lake to substantiate them.

Step 3 (Precipitation):

Crystal energy at time t $Q_i(t)$ is influenced by three factors as we have discussed.

$$Q_i(t) = \lambda_1 P_i(t) + \lambda_2 S_i(t) + \lambda_3 W_i(t).$$

Because the center of the crystal is not determined, the best position of the crystal will be regarded as the temporal center L_{center}. Similarly, the expression of influence from the center and wind is defined by

$$P_i(t) = \frac{\alpha \mathbb{P}}{||L_i - L_{center}||}$$

$$W_i(t) = \gamma \mathbb{C}.$$

Step 4 (Precipitation):

After being tested by the energy function $f(x)$, top N_p crystals will precipitate. According to the property of crystals, they will absorb energy from other crystals nearby. Therefore, a crystal shell with extension boundaries is formed by them as follows.

Assuming that we have a set of top-position crystals $C_{top} = \{c_1, \ldots, c_{n+1}\}$, the freezing range of the jth dimension is

$$c_{ij\_down} = c_{\min} - |c_{\max} - c_{\min}|$$

$$c_{ij\_up} = c_{\max} + |c_{\max} - c_{\min}|$$

where $c_{\max}$ is the upper boundary of C_{top} and $c_{\min}$ is the lower boundary of C_{top} in the jth dimension.

Step 5 (Growth): After precipitation of C_{top}, an expanded crystal shell is built. Affected by the crystal shell formed in step 4, the crystals inside it will begin to precipitate and link together to grow. Thus, N_G crystals will precipitate inside the shell randomly. Test these crystals by the energy function and define their positions in the lake. However, not all the crystals will link together because of the energy absorbed from the center. Tested by the energy function, the top N_p crystals will be reserved for the next iteration, while others will link to the crystal shell.

Step 6: By the end of step 5, there will still be S crystals in the lake. Repeat step 1 to step 5 for T iterations.

A2. Experimental Results

The set of benchmark functions contains eight functions that are commonly used in evolutionary computation literatures to show the solution quality and convergence rate

(Zhang, Zhu, and Chen 2014). The first four functions are unimodal problems, and the remaining functions are multimodal. The functions are listed in Table A1, including their dimensions, initialization ranges, and global optimum. Parameter settings of CEO are listed in Table A2.

Results of the mean value and standard deviation for 30 and 45 dimensions are listed in Table A3. Each of the experiments in this section is repeated 50 times. The experimental results of CEO are compared with the results obtained by root growth algorithm and PSO of Zhang et al. (2014). For the eight benchmark functions, we have much better solution than them in both mean value and standard deviation. Therefore, the effectiveness of the CEO on continuous optimization has been verified.

TABLE 0A1. Benchmark Functions.

Benchmark	Function	Dimensions	Range	Minimum
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	30 and 45	$[-100, 100]^D$	0
SumSquares	$f_2(x) = \sum_{i=1}^{D} i x_i^2$	30 and 45	$[-10, 10]^D$	0
Rosenbrock	$f_3(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	30 and 45	$[-30, 30]^D$	0
Schwefel2.22	$f_4(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30 and 45	$[-10, 10]^D$	0
Rastrigin	$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30 and 45	$[-10, 10]^D$	0
Schwefel	$f_6(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30 and 45	$[-500, 500]^D$	−12,569.5
Ackley	$f_7(x) = 20 + e - 20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)} - e^{\left(\frac{1}{D}\sqrt{\sum_{i=1}^{D}\cos(2\pi x_i)}\right)}$	30 and 45	$[-32.768, 32.768]^D$	0
Griewank	$f_8(x) = \frac{1}{4000}\sum_{i=1}^{D} \lvert x_i^2 \rvert - \left(\prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1$	30 and 45	$[-600, 600]^D$	0

TABLE 0A2. Parameter Settings of the Crystal Energy Optimizer.

Parameters	Value
Number of crystals initialized	100
Initial energy E_{i0}	$\prod_{j=1}^{n} \lvert Range_{ij} \rvert$
Number of dimensions n	30 and 45
Number of crystals forming shell N_p	$n + 1$
Number of crystals precipitate inside shell N_s	80
Scale factors α, β, γ	[0, 1]
Decision factor for center energy λ_1	100 (process of precipitation); 0 (process of growth)
Decision factor for frozen crystal λ_2	0 (process of precipitation); −100 (process of growth)
Decision factor for wind λ_3	−50

TABLE 0A3. Comparison of CEO with PSO and GRA on Benchmark Functions.

	Dimension					
	30D			45D		
	PSO	RGA	CEO	PSO	RGA	CEO
	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)
Sphere	2.25409E−008	1.09852E−008	**9.76E−29**	2.27609E−006	1.93873E−006	**4.11635E−15**
	1.76046E−008	1.61596E−008	**1.56979E−44**	2.62744E−007	1.22332E−006	**4.3402E−19**
SumSquares	1.15988E−003	4.07359E−003	**1.62E−28**	7.25119E−006	9.84732E−006	**1.76248E−14**
	6.23641E−004	5.53186E−004	**1.61797E−28**	6.5067E−006	2.94736E−006	**3.18165E−24**
Rosenbrock	36.0147	21.2087	**9.053401291**	41.0732	34.3239	**0.19875512**
	37.1823	**0.808404**	7.2691621	3.8414	2.11109	**1.450063024**
Schwefel2.22	0.0958865	0.0549662	**9.37485E-04**	0.00681453	0.00268347	**2.05249E−05**
	0.0709508	1.99853E−003	**7.53279E−06**	0.000966737	0.000429901	**9.69806E−07**
Rastrigin	28.5221	4.28574E−004	**3.113E−05**	74.6218	0.55754	**0.21239**
	9.02802	9.69091E−005	**4.23953E−05**	4.33688	**0.943137**	1.21301
Schwefel	−5,251.66	−8,487.84	**−13,000.77703**	−9,864.92	−11,494.2	**−13,124.98124**
	1,176.51	**1,041.19**	1,705.099791	1,768.21	1,658.22	**1,645.227447**
Ackley	6.44018E−007	3.27635E−004	**5.2453E−005**	0.00346664	0.00806774	**0.01253**
	4.68484E−007	1.28737E−005	**5.141879E−009**	0.00164182	0.00285592	**0.2411764E−404**
Griewank	0.0338252	3.70074E−003	**1.11798E−11**	0.0249269	0.00579939	**0.002457329**
	0.0483038	5.22626E−003	**4.05804E−20**	0.042542	0.0050909	**1.34077E−06**

CEO, crystal energy optimizer; PSO, particle swarm optimizer; RGA, root growth algorithm. The optimal mean and standard value of each instance have been highlighted in the table.