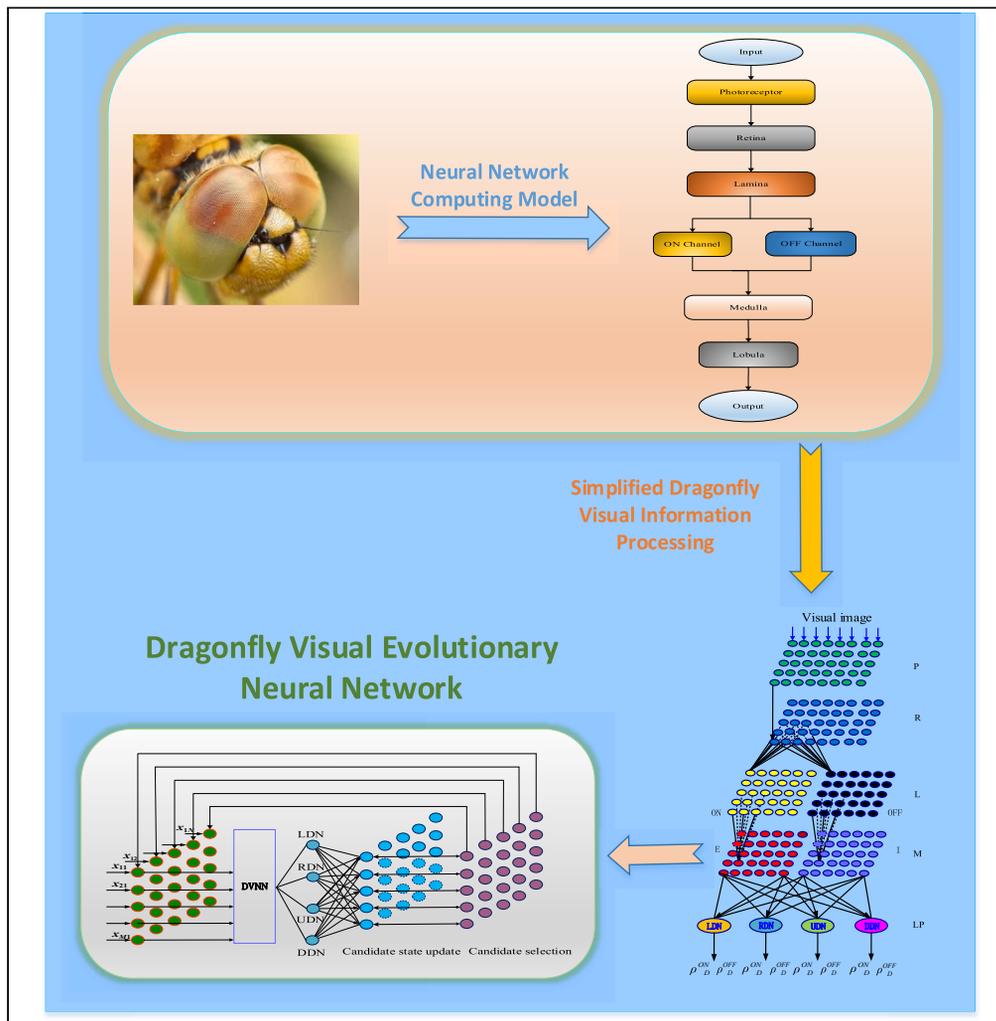


Article

Dragonfly visual evolutionary neural network: A novel bionic optimizer with related LSGO and engineering design optimization



Heng Wang,
Zhuhong Zhang

zhzhang@gzu.edu.cn

Highlights
Developing a new-type visual evolutionary neural network for LSGO problems

A dragonfly visual neural network is suggested to guide the state's transition

A differential evolution-like strategy updates the current states

An operational amplifier is optimized by the evolutionary neural network

Wang & Zhang, iScience 27, 109040
March 15, 2024 © 2024 The Author(s).
<https://doi.org/10.1016/j.isci.2024.109040>



Article

Dragonfly visual evolutionary neural network: A novel bionic optimizer with related LSGO and engineering design optimization

Heng Wang^{1,3} and Zhuhong Zhang^{1,2,4,*}

SUMMARY

Biological visual systems intrinsically include multiple kinds of motion-sensitive neurons. Some of them have been successfully used to construct neural computational models for problem-specific engineering applications such as motion detection, object tracking, etc. Nevertheless, it remains unclear how these neurons' response mechanisms can be contributed to the topic of optimization. Hereby, the dragonfly's visual response mechanism is integrated with the inspiration of swarm evolution to develop a dragonfly visual evolutionary neural network for large-scale global optimization (LSGO) problems. Therein, a gray-scale image input-based dragonfly visual neural network online outputs multiple global learning rates, and later, such learning rates guide a population evolution-like state update strategy to seek the global optimum. The comparative experiments show that the neural network is a competitive optimizer capable of effectively solving LSGO benchmark suites with 2000 dimensions per example and the design of an operational amplifier.

INTRODUCTION

With the rapid development of big data and artificial intelligence, various engineering design optimization problems with unknown large-scale parameters increase exponentially, e.g., Li-ion battery health prediction,¹ offline time-sensitive load scheduling,² railway passenger network operation,³ optimal clustering design,⁴ remote sensing images,⁵ electronic system design,⁶ deep neural networks,⁷ diagnosis of COVID-19,⁸ community acquired pneumonia,⁹ the second-order perturbation delay Lane–Emden model in astrophysics,¹⁰ the fifth type of induction motor model,¹¹ etc. Even though such problems can be well expressed by LSGO models, they will face a great challenge when solving their optima due to complex variables interaction or hard constraints. The main difficulty involves three points: (1) when the number of decision variables enlarges, the search space will expand increasingly, which necessarily makes it extremely difficult to seek the optimal solution because of extensive time consumption and the high requirement of optimization ability; (2) lots of LSGO problems usually share the property of either multimodality or pathogenicity, and hence their global optima can hardly be found in that usually there exist a great number of local optima around the global optimal solution; and (3) if an LSGO problem itself is a black box, the correlation between variables is unknown, and hence the comparison between candidate solutions gets into trouble.

Whereas some traditional numerical methods such as the Newton's method¹² and conjugate gradient method¹³ can handle continuously differentiable convex optimization problems with large-scale decision variables, their convergence speed and accuracy are desired to be improved greatly. On the other hand, even though metaheuristic approaches such as CSO,¹⁴ GPBO,¹⁵ JOA,¹⁶ RBLSO,¹⁷ PSO,¹⁸ ACO,¹⁹ and NPSO²⁰ with strong competition in the metaheuristics branch can rapidly handle multiple kinds of continuous or non-continuous optimization problems, they, with usually fixed parameter settings, are easily trapped into local optima when confronted with LSGO problems. For example, CSO, a cooperation-based hybrid particle swarm optimization approach, has the advantages of strong local search ability, few parameters, and wide applicability, but its performance rapidly deteriorates with increasing dimensionality. Consequently, it is still desired to explore new-type and effective approaches for seeking the LSGO problem's global optima efficiently. These indicate that new computational models, inspirations from nature, and approaches are urgently needed to cope with LSGO problems. Interestingly, when exploring interdisciplinary optimization approaches, maybe it is a unique inspiration that the visual neural network's activity output online guides the swarm evolution-based state transition in order to excavate new-type and high-performance optimization approaches.

Biological visual systems have the natural abilities of rapid and prompt response, feature selection, noise suppression, direction recognition, etc., which can provide us with rich inspirations for constructing computational models to tackle specific engineering problems. Particularly, bio-inspired feedforward visual neural networks as a special type of neural network, which originate from biological visual information-processing

¹College of Big Data and Information Engineering, Guizhou University, Guiyang, Guizhou 550025, P.R. China

²Guizhou Provincial Characteristic Key Laboratory of System Optimization and Scientific Computing, Guiyang, Guizhou 550025, P.R. China

³Tongren Polytechnic College, Tongren, Guizhou 554300, P.R. China

⁴Lead contact

*Correspondence: zhzhong@gzu.edu.cn

<https://doi.org/10.1016/j.isci.2024.109040>



systems, can well exhibit some significant visual response characteristics and have unique advantages over conventional computation models in handling visual image-based engineering application examples such as object tracking, collision detection, etc. Especially, the dragonfly visual system is highly sensitive to visual motion changes of moving objects, and can thus rapidly track small moving objects in visual scenes. As a unique type of insect, it can emerge visual guidance behaviors by its distinctive, rapid, and high-sensitive visual neural information-processing layers. These can provide researchers with unique inspirations in constructing visual neural networks with structural simplicity. To our best knowledge, whereas some neuron-based neural circuit models were discovered in neurophysiology and applied to object recognition, there has been no any reported dragonfly visual neural network in the literature. Therefore, it still keeps open to develop dragonfly visual computational models and explore their potential engineering applications, in particular LSGO. This motivates us to probe into a new research topic- dragonfly visual neural networks. Hereby, the current work tries to develop a dragonfly visual neural network and integrates it with the differential evolution-like inspiration to develop a dragonfly visual evolutionary neural network (DVENN) between artificial intelligence and computer vision, while probing into its potential application value to LSGO and operational amplifier design optimization in the field of the integrated circuit. This can not only help to reveal the functional characteristics of the dragonfly visual system from the perspective of computer simulation, but also provide an alternative optimizer for complex problems.

It is highlighted that by breaking through the shackle of conventional intelligent optimization research, the current work concentrates on discussing both the possibility of combining dragonfly visual neural networks with the inspiration of swarm evolution, while developing a fast visual evolutionary neural network to open a prompt pathway to effectively solving LSGO problems, in particular the problem of operational amplifier design in the field of the integrated circuit. The main innovations are summarized below.

- (1) Inspired by the visual hierarchical information processing mechanism and the response characteristics of four motion-sensitive neurons (i.e., the left, right, upper and down motion direction detection neurons) in the dragonfly visual system, a feedforward dragonfly visual neural network (DVNN) with four subnetworks is, *for the first time*, developed to characterize changes in visual motion, being capable of online outputting motion direction activities named learning rates. Each subnetwork generates a pair of on-line learning rates along the left, right, upper or down direction to guide each candidate solution/state move toward the global optimal solution.
- (2) Based on the inspiration of swarm evolution and the unique physiological characteristics of the above four neurons, a new-type state update model is proposed to update the current state matrix into a more valuable one. Therein, each given state derives four candidate states, based on a differential evolution-like state transition strategy and the four above-mentioned learning rates, after which the state is updated by the best of the candidate states.
- (3) Related to (i) and (ii), DVENN, which completely differs from any existing optimizers, is developed to solve LSGO problems, particularly pathological function optimization examples. Thereafter, based on the design principle and analysis of the analog integrated circuit, a two-stage open-loop operational amplifier design problem is skillfully solved by DVENN, after being modeled into a strongly nonlinear constrained optimization model.

Unlike any existing optimization approaches, DVENN is a state matrix transition-based optimization approach with only the unique parameter of input resolution. Therein, DVNN can online regulate the parameters' setting in the state update scheme such that the elements in the current state matrix seek the optimal solution omnidirectionally. Thereby, it plays an import role in guiding the current state's transition and regulating the state transition's parameters. On the other hand, DVENN, which possesses the ability of strong exploration and exploitation, can effectively LSGO problems efficiently.

Related research work

Although some state-of-the-art optimization algorithms are competitive for high-dimensional optimization problems, they often encounter a great challenge when faced with multi-modal LSGO problems. The two mainstream research branches, i.e., traditional numerical optimization and intelligent optimization, have been contributed to the topic of LSGO.

Traditional numerical optimization

As a classical type of deterministic iterative approach, numerical optimization plays an important role in solving LSGO problems due to the striking characteristics of single-state transition, gradient descent, and precise iteration, in which gradient information is usually used to guide the current state to transfer toward the optimal solution. However, even though lots of theoretical analyses and experiments have validated that numerical optimization approaches are alternative and rapid in solving LSGO problems, their effectiveness and solution accuracy are still unsatisfactory. For example, to enhance the rate of convergence and the solution's accuracy, Tang et al.²¹ developed an asynchronous parallel algorithm to solve the large-scale composition optimization problems with non-smooth regularization penalty terms, depending on the asynchronous parallel implementation and variance reduction. Besides, Liu et al.²² proposed an efficient parameter-level parallel optimization algorithm after transforming large-scale spatiotemporal data mining into optimization models. Therein, they employed the extrapolation strategy to raise the convergence rate of gradient descent. Further, Maleknia et al.²³ constructed a gradient sampling scheme, and later, an improved stochastic gradient descent approach was derived to handle non-smooth nonconvex optimization problems. More recently, Xu et al.²⁴ developed a gradient approximation approach, in which the gradient descent direction was determined by several representative gradients.

Intelligent optimization

Compared to numerical iterative optimization, intelligent optimization has emerged a strong momentum of development due to low problem-specific dependence, strong solution search performance, convergence quickness, etc. Some recent co-evolutionary approaches indicated that evolutionary computation is an irreplaceable mainstream branch in discussing how LSGO can be well solved. Meanwhile, several state-of-the-art co-evolutionary algorithms are speeding up the development of evolutionary computation toward probing into competitive LSGO approaches. Besides, researchers have also paid great attention to concerning how metaheuristics approaches get rid of the shackle of being trapped in local optima when tackling complicated LSGO problems. The reported co-evolutionary or metaheuristic approaches are valuable when faced with decomposable, semi-decomposable, or convex LSGO problems. However, once they are used to solve indecomposable or strongly nonlinear problems, their disadvantages, e.g., local convergence, will timely expose thoroughly due to the curse of dimension and the increasing complexity of practical problems. In other words, as the dimension increases, their performance will deteriorate seriously. Thereby, new, and competitive intelligent approaches are desired to promote the ability to seek the optimal solution. The research progress of LSGO-based intelligent optimization can be summarized below.

Evolutionary optimization

Co-evolution as a type of divide-and-conquer approach, originally proposed by Potter et al.²⁵ always dominates the research progress of LSGO-based evolutionary algorithms. It decomposes an LSGO problem into a series of small-scale optimization problems and ultimately obtains the optimal solution through the co-evolution of sub-populations. Herein, the strategies of problem-driven variable decomposition and sub-population communication play a crucial role in determining whether evolutionary algorithms can acquire the optimal solution. The existing variable decomposition strategies can be classified into three broader classes, i.e., fixed grouping,^{25,26} random grouping,^{27,28} and dynamic grouping.^{29–31}

Usually, once the correlation between variables is simply taken into consideration, the coupling ability between evolving sub-populations become weak, and hence the performance of solution search is influenced seriously. Some fixed grouping strategies, e.g., folding grouping, can simply divide decision variables into several sub-groups, in which the correlation of variables is simply considered by dimension compromise. They, together with genetic operators, can well solve separable or semi-separable LSGO problems. However, when the correlation between coupled variables become stronger, such a type of grouping strategy cannot help to improve the performance of solution search. To break through the shackle, Jia et al.²⁶ devoted a contribution-based cooperative co-evolutionary algorithm to decompose and optimize inseparable LSGO problems with overlapping subcomponents, in which the grouping strategy had an important impact on solution search quality.

Recently, several variable grouping strategies, e.g., random grouping, dynamic grouping, etc., have frequently been exploited to decompose the decision variables of the optimization problem into multiple sub-blocks with the same or varying sizes during the evolution process. Regarding the application of the random grouping strategy, Kaboli et al.²⁷ developed an artificial cooperative search algorithm to accurately solve the economic dispatch problem in the precondition of less complexity, while Sun et al.²⁸ acquired a hybrid cooperative co-evolutionary algorithm in which the Markov random field decomposition strategy was used to execute variables decomposition. Nevertheless, one such decomposition strategy easily causes the degradation of the solution search performance.

The strategy of dynamic grouping can not only adjust the size of each variable sub-group dynamically and effectively, but also overcome the premature phenomenon incurred by improper dimension decomposition in the process of co-evolution. For example, Mahdavi et al.²⁹ developed an incremental grouping method to merge sub-groups in the process of sub-problem optimization, which could make full use of limited resources to solve many more sub-problems. Besides, a cooperative co-evolutionary differential evolution approach with high efficiency, developed by Wang et al.³⁰ can solve large-scale numerical optimization problems without prior correlation between variables. Yin et al.³¹ designed a multi-population dynamic coevolutionary strategy able to adjust the connection dynamically and adaptively between individuals. Nevertheless, such a grouping strategy might incur computationally expensive costs and non-uniform information distribution as well.

Metaheuristic optimization

Metaheuristics as a type of special swarm intelligence approach have initially touched upon the research of LSGO in recent years because of their simplicity and low computational complexity. However, since they easily encounter the premature phenomenon, multiple metaheuristic search strategies are usually integrated to adapt to different kinds of LSGO problems, in which collaborative mechanisms are used to activate the interaction between the search strategies in population exploration and exploitation. For example, Cuevas et al.³² developed two types of metaheuristic approaches with strong competitiveness-social spider optimization (SSO) and material state search (SMS) for solving complex optimization problems, based on the inspirations of swarm evolution and the simulation of the state of material phenomena. Luo et al.³³ combined a collaborative mechanism with the simulated annealing algorithm to develop a hybrid optimization approach in which a probability-based local search chain was used to promote the abilities of global exploration and centralized search. Ma et al.³⁴ developed an improved approach after analyzing that differential evolution approaches were easily trapped into a local optimum due to the similarity between individuals at the later stage of evolution. Herein, the current population was randomly divided into three sub-populations with the same sizes, each of which adopted a unique mutation strategy to generate new individuals. On the other hand, some other hybrid metaheuristic approaches, e.g., gray wolf and wild goose optimization ones, have preliminarily touched upon the topic of LSGO. For example, inspired by a school of hunting sailfish, Shadravan et al.³⁵ exploited a novel nature-inspired metaheuristic algorithm in which the sailfish population

was employed to intensify the ability to seek the optimal solution, and the sardine population helped to strengthen the diversity of solution search. Sabir et al.³⁶ proposed a neuro swarm computational heuristic model which integrated neural networks with particle swarm optimization to solve nonlinear coupled Emden–Fowler models. At the same time, they³⁷ combined genetic algorithm (GA) with sequential quadratic programming (SQP) to optimize Goodman neural networks (GNNs) used for solving a class of singular periodic nonlinear differential systems (SP-NDS) in nuclear physics. Recently, Fu et al.³⁸ constructed a game model to simulate the ownership and dissemination of node information driven by interests, and proposed a competitive particle swarm optimization algorithm based on de-heterogeneous information by simulating the spread of competitive influence.

Artificial visual computational model

Biological visual systems as a natural resource have played an important role in solving lots of engineering problems, such as car collision detection and robot navigation, since some striking visual characteristics can be dedicated to exploring new-type artificial computation models. Basically, many insects with compound eyes can quickly detect the position and direction of a moving object in terms of their visual response mechanisms. Particularly, dragonflies have unique visual information processing mechanisms which can provide us with rich inspirations for constructing artificial visual neural computation models.

Dragonfly visual neural computation models

The dragonfly visual system, which can detect changes in visual motion, is very sensitive to visual motion in the field of view. It can not only estimate the distance between moving objects, but also identify and track the status of visual scenes. These unique visual characteristics have prompted researchers to perform lots of studies about the dragonfly's visual response mechanism. Unfortunately, no neurophysiologic finding on such a response mechanism has been borrowed to construct any dragonfly visual neural network capable of serving comprehensive engineering applications.

In the visual system, there exists a widely studied class of motion-sensitive neurons called small targets motion detectors (STMDs) with unique target motion detection functions, each of which can respond strongly to a moving small target.³⁹ To detect the behavior changes of a small target, Wiederman et al.⁴⁰ constructed an elementary small target motion detector (ESTMD) model morphologically. However, no computation models can achieve the STMD's motion direction detection. Fortunately Hassenstein et al.⁴¹ and Reichardt et al.⁴² pioneered an elementary motion detector (EMD) capable of detecting the unidirectional movement direction between two visual points. Subsequently, to realize bidirectional motion direction detection between two visual nodes, Egelhaaf et al.⁴³ developed a correlative detector formed by two symmetrically arranged EMD models. These provide researchers with important foundations in exploring computational models of dragonfly visual motion direction detection. Based on these basics, Bagheri et al.⁴⁴ constructed two computational models to simulate the dragonfly's characteristics of motion direction detection after integrating EMD and ESTMD, i.e., ESTMD-EMD and EMD-ESTMD. The two hybrid models have been successfully used to track autonomous mobile ground robots in cluttered backgrounds. Recently, Wang et al.⁴⁵ constructed a direction-selective STMD model able to provide a unified and rigorous mathematical description by comparison against other STMD models.

Related visual computation model

Studies on biological visual neural networks in motion detection and object tracking originate from the above-mentioned EMD motion direction detector. Based on EMD, Ögmen et al.⁴⁶ designed two fly visual neural models to perceive the motion cues of moving objects. Subsequently, Missler et al.⁴⁷ developed a five-layer visual neural network to track a small object in a black-and-white background by means of the internal structure of the fly visual lobe. Based on these works, Zhang et al.⁴⁸ developed a fly visual neural network to demonstrate the functional response characteristics of the fly's interneurons and perform car collision detection. To solve the problem of motion tracking and navigation, Fu et al.⁴⁹ constructed a hybrid visual neural network to detect visual collisions. Besides, Wang et al.⁵⁰ designed a neural computation model to filter out irrelevant motion signals in the background of clutter related to the functions of a lobular plate tangential cell (LPTC), a two-quadrant motion detector, and background motion direction estimation.

On the other hand, Rind and Bramwell et al.⁵¹ used the locust visual nervous system's response mechanism to construct a locust visual neural network composed of perception, excitation, inhibition, and integration layers as well as an LGMD. Subsequently, based on their research work, Yue et al.⁵² used the characteristics of the LGMD to intensively discuss the problems of object motion pattern recognition and collision detection, and later, obtained a series of locust visual neural networks. In recent years, Hu et al.⁵³ proposed an artificial locust visual neural network model suitable for such movement patterns recognition as translation and rotation, based on the visual response characteristics of locusts and macaques. Related to the visual response characteristics of the fruit flies and locusts, Fu et al.⁵⁴ proposed a hybrid neural network model to implement motion tracking and robot navigation.

Problem background and motivation

The general continuous-type large-scale global optimization (LSGO) problem is expressed by

$$\min f(\mathbf{x}) = f(x_1, x_2, \dots, x_D), 1 \leq i \leq D \quad (\text{Equation 1})$$

where $\mathbf{x} \in \Omega$ is called a state throughout the current work; D denotes the variable dimension with $D \geq 1000$; and Ω denotes the bounded decision domain in R^D .

Definition 3.1 Local optimal solution: \mathbf{x}^* in Ω is said to be a local optimal solution, if there exists δ such that, when $\|\mathbf{x} - \mathbf{x}^*\| < \delta$, \mathbf{x}^* satisfies $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

Definition 3.2 Global optimal solution: $\mathbf{x}^* \in \Omega$ is said to be the global optimal solution, if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for $\forall \mathbf{x} \in \Omega$.

Usually, the characteristic of the function $f(\mathbf{x})$ determines the difficulty of solving the above LSGO problem. If $f(\mathbf{x})$ is the accumulation of several sub-objective functions, the optimal solution can be found by parallel processing. However, if the decision components in the decision vector are strongly correlated, it is difficult to find the solution of the problem even if the divide-and-conquer approach is integrated with evolutionary approaches. Particularly, once the LSGO problem is multimodal or ultra-high dimensional, any existing approaches, e.g., numerical optimization ones, become extremely difficult in seeking the global solution(s). Summarily, when tackling LSGO problems, three conventional issues need to be considered: (1) how the curse of dimensionality influences the efficiency and quality of solution search; (2) how the capabilities of global exploration and local exploitation can be enhanced without prior knowledge; and (3) how the strategies of evolution or state update can excavate valuable solutions with minimal time consumption.

Like other insect visual systems, the dragonfly visual system also possesses five sequentially connected neural information processing layers, each of which processes the neural signals from the last layer by a specific visual mechanism. In each neural layer, each node collects the neural signal of the counterpart in the last layer at each moment or those of the nodes around the counterpart, and later, transmits a neural signal to the counterpart in the next neural layer. One such response process can generate global motion direction activities to characterize the global motion direction change of visual movement. Based on such an inspiration, the dragonfly's visual response process is simplified to construct a feedforward dragonfly visual neural network (DVNN). Further, based on an analogy to the process of population evolution, such a neural network is integrated with a completely new and differential evolution-like state update strategy to constitute DVENN able to handle LSGO problems. Herein, the leftward, rightward, upward, and downward global motion direction activities acquired by DVNN at each moment are taken to guide the states' transition in the process of solution search.

Hereby, take any $M \times N$ states to form an abstract state matrix \mathbf{X} , i.e., $(\mathbf{x}_{ij})_{M \times N}$. Each state \mathbf{x}_{ij} is evaluated by the related object value $f(\mathbf{x}_{ij})$ regarded as a gray value here. Such $M \times N$ gray values constitute a gray image with resolution $M \times N$, i.e., $f(\mathbf{X})$. Let $\mathbf{X}^{(t)}$ and $f(\mathbf{X}^{(t)})$ be the state matrix and the related gray image at the t -th moment, respectively. The state (i, j) in $\mathbf{X}^{(t)}$ is denoted by $\mathbf{x}_{ij}^{(t)}$. Accordingly, the current work tries to develop a state matrix transition-based dragonfly visual evolutionary neural network (DVENN) to seek the global optimal solution of the above LSGO problem. Herein, $\mathbf{X}^{(t)}$ is updated by $\mathbf{X}^{(t+1)}$, namely

$$\mathbf{x}_{ij}^{(t+1)} = \mathbf{x}_{ij}^{(t)} + \Delta \left(\mathbf{x}_{ij}^{(t)}, \rho^{ON}, \rho^{OFF}, \mathbf{x}_{gb} \right) \quad (\text{Equation 2})$$

with $1 \leq i \leq M, 1 \leq j \leq N$, where Δ is the transition increment vector of the state $\mathbf{x}_{ij}^{(t)}$; \mathbf{x}_{gb} is the best state found until now.

Related to the state matrices $\mathbf{X}^{(t-1)}$ and $\mathbf{X}^{(t)}$, the visual neural network DVNN outputs four pairs of ON-OFF motion direction activities at the t -th moment along the leftward, rightward, upward, and downward visual motion directions, i.e., $(\rho_L^{ON}, \rho_L^{OFF}), (\rho_R^{ON}, \rho_R^{OFF}), (\rho_U^{ON}, \rho_U^{OFF}),$ and $(\rho_D^{ON}, \rho_D^{OFF})$, by which (ρ^{ON}, ρ^{OFF}) is taken as the best of the four pairs based on a specific evaluation rule. The state update strategy in (Equation 2) is designed based on the inspiration of differential evolution, where the main contribution of the pair of ρ^{ON} and ρ^{OFF} is to purposely guide each current state to transfer toward the region on which the optimal solution exists. To be able to find the solution as fast as possible, the current concerns incorporate two points: (1) based on the dragonfly visual response mechanism, how DVNN can be constructed to generate four pairs of global motion direction activities at each moment such that the quality of state transition can be enhanced, and (2) how the inspiration of differential evolution can be used to construct a state matrix-based state update strategy such that each state is transformed into another high-quality state with the guidance of a pair of motion direction activities. These concerns will be solved in the following sections.

Numerical experiment

This section firstly formulates the environmental setting. Secondly, the characteristics of the CEC'2010 and 2013 suites^{55,56} are analyzed. Finally, the experimental results and analyses are included in the last section.

Experimental environment and benchmark suite

Throughout the whole experimental study below, all experiments are executed on a computer with Intel Core (TM)i5-9400F CPU/2.90GHz/RBM/16.0 GB/Visual C++ platform. Seven state-of-the-art metaheuristics, namely CSO,¹⁴ GPBO,¹⁵ JOA,¹⁶ RBLSO,¹⁷ PSO,¹⁸ ACO,¹⁹ and NPSO,²⁰ are selected to participate in comparison with DVENN. Each method terminates its solution search when the total of fitness evaluations exceeds 3×10^6 , while executing twenty-five times independently for each example. The parameters setting of each compared approach originates from the related literature, while DVENN takes $M \times N$ as 8×8 . The thirty-five test examples with dimension 1000 or 2000 per one are taken from the CEC'2010 and 2013 benchmark suites to examine whether DVENN outperforms the chosen metaheuristics. Their characteristics are summarized in Table 1.

The CEC'2010 benchmark suite with dimension 2000 per example as a hard LSGO test set contains twenty continuously differential examples whose characteristics involve separability, partial separability, complete inseparability, variable rotation, variable overlapping, etc. Even though the eight examples, i.e. $f_1, f_4, f_7, f_9, f_{12}, f_{14}, f_{17},$ and f_{19} in the benchmark suite, are unimodal, it is still difficult for any existing

Table 1. Characteristics of each benchmark example with $D = 2000$

Test suite	Example	Type	Characteristic	Optimum
CEC'2010	$f_1 - f_4$	Separable	Uni- or Multi-modality	0
	$f_5 - f_6$	Single-group m-rotated and non-separable	Uni- or Multi-modality	0
	$f_7 - f_8$	Single-group m-non-separable	Uni- or Multi-modality	0
	$f_9 - f_{14}$	$D/2m$ -group m-rotated and non-separable	Uni- or Multi-modality	0
	$f_{15} - f_{16}$	D/m -group m-rotated and non-separable	Uni- or Multi-modality	0
	$f_{17} - f_{18}$	-group m- non-separable	Uni- or Multi-modality	0
	$f_{19} - f_{20}$	Fully non-separable	Uni- or Multi-modality	0
CEC'2013	$f_1 - f_3$	Separable	Uni- or Multi-modality	0
	$f_4 - f_{10}$	Partially separable, rotation	Uni- or Multi-modality	0
	$f_{11} - f_{14}$	Partially separable, rotation, overlapping	Uni- or Multi-modality	0
	f_{15}	Fully non-separable	Uni- or Multi-modality	0

metaheuristic method to quickly obtain their global solutions due to large search spaces and complex objective functions. The CEC'2013 benchmark suite with dimension 1000 per example, which involves fifteen examples with complicated objective functions, is more difficult in seeking the examples' optima.

Experimental results and analysis

Case I: The CEC's 2010 benchmark suite

Each of the above eight methods performs twenty-five single runs for each test example, in which the minimal objective of each example is 0. After that, their statistical results, displayed in Table 2 below are used to perform the approaches' performance analyses. In addition, the t-test with the significance level of 5% is here chosen to test whether DVENN can provide a significant improvement over each compared method, after which the acquired results are given at the bottom of the table. Besides, we take f_5 , f_{10} , f_{15} , and f_{20} as examples to plot the approaches' average search curves shown in Figure 1 below.

The results in the upper half of Table 2 demonstrate that, when coping with the above large-scale, high-dimensional, and strongly nonlinear optimization problems, the eight approaches emerge significantly different optimization capabilities. DVENN can successfully and stably acquire the theoretical or approximate optima for fifteen of twenty examples. Nevertheless, PSO and ACO are trapped into the local optima and cannot acquire the global optimum for any of the twenty examples;

GPBO, JOA, and BPSO can obtain the approximate solutions for one or two examples; RBLSO and CSO can only gain the approximate solutions for five examples. These illustrate that DVENN has the abilities of strongly global exploration and local exploitation, since its DVNN can purposely guide the current states to transfer toward the optimal solution(s) along specific directions with the help of multiple kinds of learning rates generated by it. Thereby, it is an important inspiration to regulate the direction of state transition by means of feedforward visual neural networks. However, the compared approaches usually include nonadaptive parameters and update their states by virtue of the guidance of the current best individual, and hence easily get stuck into local search at a high probability.

On the other hand, the approaches' performance characteristics are compared in terms of optimization quality and search stability. The t-test based results in the bottom as in Table 2 show that DVENN significantly outperforms the compared approaches in that it can defeat PSO, ACO, GPBO, JOA, RBLSO, NPSO, and CSO for 20, 20, 20, 19, 14, 14, and 12 examples, respectively. NPSO is superior to each of the other compared approaches, but easily gets stuck into a local optimum. Thereby, DVENN has the strong ability to handle different kinds of LSGO problems, but the compared approaches, e.g., PSO, ACO, GPBO, JOA, and BPSO, need to make great improvements so that their capabilities of global exploration and local exploitation can be balanced properly.

The average search curves in Figure 1 verify that DVENN can take a rational tradeoff between global exploration and local exploitation. It can adaptively regulate the runtime of global exploration and local exploitation. In other words, when solving a relatively easy example, DVENN needs less time to execute global exploration, and conversely, it takes much more time to comprehensively excavate those potential solutions. Therein, DVNN plays an important role in regulating the amplitude, scope, and direction of state transition. Furthermore, Figure 1 confirms that DVENN can converge to the global optimum for each example f_5 , f_{10} , or f_{20} , but PSO and ACO behave poorly.

Case II: The CEC's 2013 benchmark suite

To further confirm that DVENN is a competitive optimizer, it is further compared to the above seven chosen methods in terms of the test suite. Like the above experiment, they also execute twenty-five times on each example, respectively. The minimal objective value of each example is 0. Afterward, the acquired statistical results in Table 3 are utilized to perform a comparative performance analysis of the above eight methods. Similarly, only take f_3 , f_6 , f_9 , and f_{12} for example to emerge the approaches' average search curves as in Figure 2.

Table 2. Comparison of statistical results acquired by the eight approaches for the CEC'2010 benchmark suite

<i>f</i>	<i>Q</i>	RBLSO	PSO	ACO	GPBO	JOA	NPSO	CSO	DVNN
<i>f</i> ₁	μ	1.12E-20↑	2.83E+11↑	3.27E+11↑	1.85E+00↑	3.95E-10↑	4.06E-21↑	5.19E-17↑	0.00E+00
	σ	5.16E-22	8.47E+10	1.52E+10	3.82E-02	3.02E-11	3.66E-22	9.85E-18	0.00E+00
<i>f</i> ₂	μ	7.95E+02↑	1.91E+04↑	2.39E+04↑	4.09E+02↑	1.09E+04↑	2.71E+03↑	4.62E+03↑	5.11e-11
	σ	4.42E+01	9.57E+03	2.76E+02	9.32E-01	8.86E+02	2.16E+02	8.96E+02	3.11e-12
<i>f</i> ₃	μ	7.60E-14↑	2.15E+01↑	2.15E+01↑	5.84E+00↑	1.99E+01↑	7.41E-14↓	4.29E-12↓	1.80E-02
	σ	5.45E-15	1.06E-02	9.08E-02	7.38E-02	2.75E-02	5.28E-15	1.44E-12	2.11e-02
<i>f</i> ₄	μ	4.59E+11↑	9.06E+14↑	3.53E+15↑	7.39E+14↑	1.98E+12↑	3.94E+11↑	1.56E+12↑	0.00E+00
	σ	7.12E+10	8.06E+14	8.09E+14	6.99E+12	2.69E+11	1.44E+11	2.37E+10	0.00E+00
<i>f</i> ₅	μ	2.79E+08↑	6.11E+08↑	8.75E+08↑	7.08E+08↑	4.26E+08↑	2.60E+07↑	3.39E+06↑	9.66E+05
	σ	9.19E+06	2.75E+07	4.56E+07	3.07E+07	1.09E+07	1.12E+07	2.14E+05	9.70E-04
<i>f</i> ₆	μ	5.41E-09↓	2.01E+07↑	2.11E+07↑	2.09E+07↑	1.97E+07↑	1.97E+01↓	8.15E-07↓	6.48E+05
	σ	1.68E-09	3.92E+05	8.26E+04	3.91E+04	1.56E+04	2.16E-02	8.41E-08	1.56E-05
<i>f</i> ₇	μ	2.30E+03↑	1.45E+11↑	7.53E+11↑	4.01E+11↑	3.44E+03↑	6.25E+01↑	7.77E+04↑	0.00E+00
	σ	7.91E+02	3.05E+10	7.08E+10	8.16E+10	5.07E+02	2.29E+01	6.89E+04	0.00E+00
<i>f</i> ₈	μ	3.19E+07↓	2.46E+16↑	1.41E+17↑	4.72E+09↑	4.35E+06↓	1.88E+07↓	3.75E+07↓	4.04E+08
	σ	2.73E+05	4.31E+15	2.88E+16	3.93E+08	1.93E+05	3.48E+05	7.21E+05	3.56E-07
<i>f</i> ₉	μ	5.03E+07↑	2.91E+11↑	3.41E+11↑	1.73E+09↑	3.63E+08↑	5.34E+07↑	6.47E+07↑	0.00E+00
	σ	2.67E+06	3.01E+10	1.49E+10	3.21E+08	2.65E+07	5.56E+05	5.75E+06	0.00E+00
<i>f</i> ₁₀	μ	7.83E+03↑	2.39E+04↑	2.39E+04↑	8.34E+03↑	1.36E+04↑	9.12E+01↑	7.83E+02↑	0.00E+00
	σ	2.71E+03	1.55E+02	1.97E+02	6.05E+01	1.37E+02	1.22E+02	8.78E+01	0.00E+00
<i>f</i> ₁₁	μ	4.69E-13↓	2.34E+02↑	2.36E+02↑	2.12E+02↑	2.24E+02↑	2.08E+01↑	2.86E-11↓	1.08E-02
	σ	1.11E-14	1.38E-01	1.55E-01	2.14E-01	1.03E+00	2.64E-01	7.71E-12	2.08E-03
<i>f</i> ₁₂	μ	5.13E+04↑	1.42E+07↑	3.33E+07↑	2.97E+06↑	2.01E+05↑	2.35E+04↑	2.50E+04↑	0.00E+00
	σ	3.32E+03	7.55E+05	5.11E+06	1.25E+05	8.00E+03	1.46E+03	7.25E+03	0.00E+00
<i>f</i> ₁₃	μ	5.97E+02↑	3.41E+12↑	3.58E+12↑	5.34E+04↑	2.74E+03↑	4.94E+02↑	6.21E+02↑	4.90E+02
	σ	1.36E+02	7.11E+11	1.02E+11	1.43E+04	8.14E+02	1.18E+01	3.05E+02	4.70E-02
<i>f</i> ₁₄	μ	1.47E+08↑	1.97E+11↑	3.87E+11↑	1.68E+09↑	1.19E+09↑	1.58E+08↑	1.90E+08↑	0.00E+00
	σ	1.00E+07	5.09E+10	1.36E+10	2.87E+07	5.31E+07	6.48E+06	2.83E+06	0.00E+00
<i>f</i> ₁₅	μ	1.03E+04↑	2.39E+04↑	2.39E+04↑	1.68E+04↑	1.43E+04↑	1.72E+01↑	9.98E+03↑	0.00E+00
	σ	6.65E+01	2.14E+02	2.63E+02	3.44E+02	8.43E+01	1.16E+02	2.24E+01	0.00E+00
<i>f</i> ₁₆	μ	6.45E-13↓	4.29E+02↑	4.29E+02↑	4.13E+02↑	3.99E+02↑	6.07E+00↑	2.04E-11↓	7.60E-05
	σ	2.50E-14	2.39E-01	1.71E-01	3.64E-01	6.49E-01	1.78E+00	1.97E-12	3.60E-05
<i>f</i> ₁₇	μ	4.06E+05↑	2.54E+07↑	7.59E+07↑	3.25E+06↑	7.78E+05↑	1.42E+05↑	3.14E+05↑	0.00E+00
	σ	3.01E+04	2.04E+06	5.04E+06	1.19E+05	1.19E+04	1.94E+04	4.36E+04	0.00E+00
<i>f</i> ₁₈	μ	1.50E+03↑	7.68E+12↑	7.62E+12↑	2.28E+05↑	1.36E+04↑	2.24E+03↓	2.39E+03↓	7.98E+03
	σ	5.62E+02	1.28E+11	2.14E+11	4.71E+04	1.73E+03	3.69E+02	4.25E+02	1.73E-03
<i>f</i> ₁₉	μ	7.25E+06↑	4.94E+07↑	1.33E+08↑	7.57E+07↑	3.29E+06↑	1.32E+06↑	8.81E+06↑	0.00E+00
	σ	1.22E+05	1.85E+07	1.42E+07	1.01E+07	2.23E+05	7.98E+04	7.75E+05	0.00E+00
<i>f</i> ₂₀	μ	1.46E+03↑	7.92E+12↑	7.79E+12↑	2.03E+04↑	1.04E+04↑	1.79E+03↑	1.19E+03↑	1.00E+03
	σ	1.35E+02	1.75E+11	2.02E+11	2.38E+03	2.97E+03	1.50E+02	3.02E+01	9.00E-02
w/t/l		14/3/3	20/0/0	20/0/0	20/0/0	19/0/1	14/3/3	12/4/4	–

↑, =, and ↓ denote that the result acquired by DVNN is superior to, equivalent to, and inferior to that gained by one compared approach, respectively. The highlighted value on mean in each row is the best of eight values, which indicates that the related approach performs well over the other approaches. The bolded values represent the optimal results in the cec2010 standard test suite.

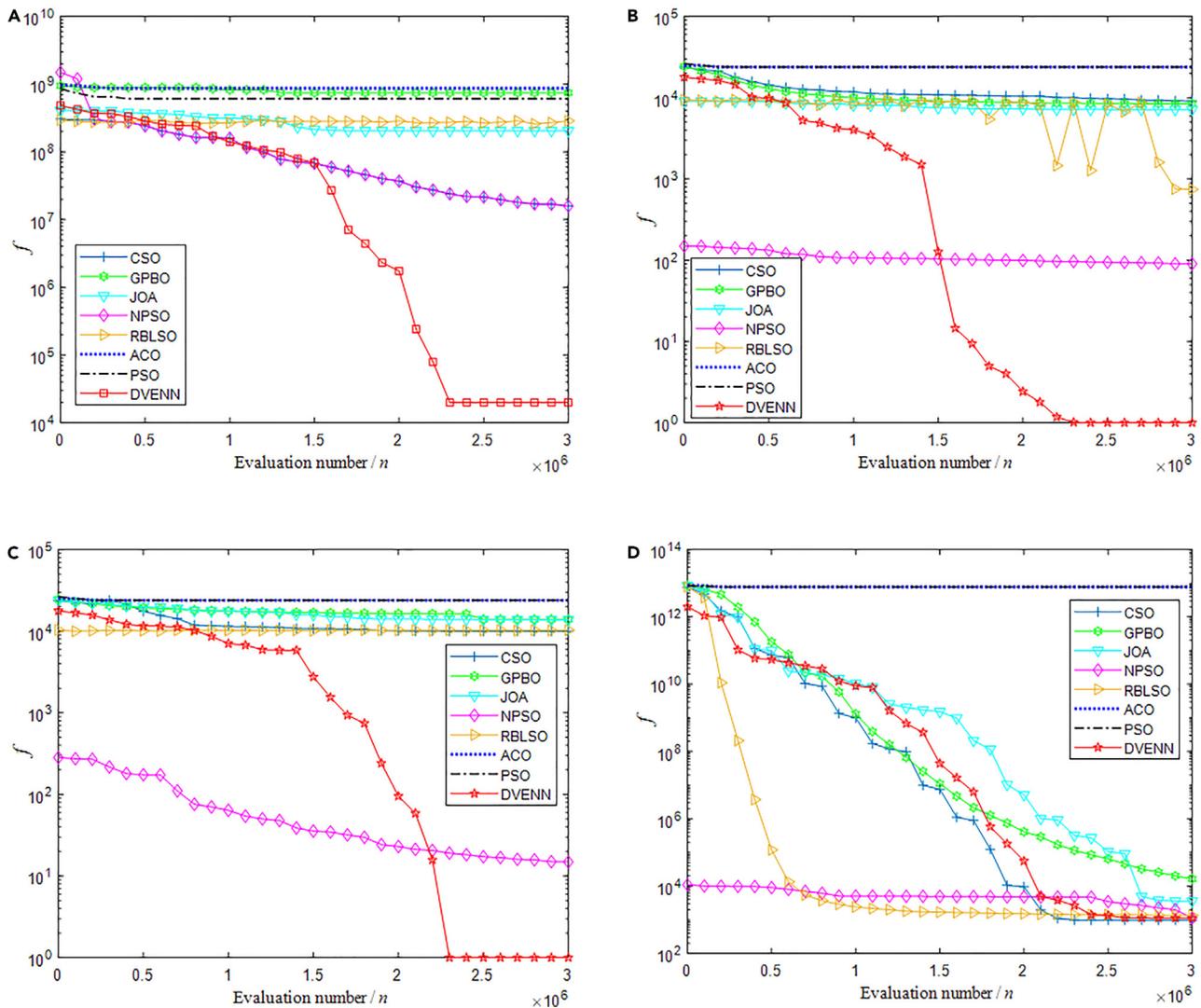


Figure 1. Comparison of the approaches' average search curves for the four examples in the CEC' 2010 suite

- (A) f_5 .
- (B) f_{10} .
- (C) f_{15} .
- (D) f_{20} .

The statistical results in the upper half of Table 3 validate that DVENN can stably acquire the optima for twelve of fifteen examples while performing well over the compared approaches for the f_{12} optimization problem even if being not capable of obtaining the problem's optimal solution. Conversely, PSO, ACO, and GPBO face a great challenge, since they cannot obtain the optimal solution for each test problem, and meanwhile, RBLSO, JOA, NPSO, and CSO can only solve one example successfully. These indicate that their diversity of population and the abilities of exploitation and exploration need to make great improvements in solving LSGO problems. On the other hand, the t-test shows that DVENN can defeat PSO, ACO, GPBO, JOA, RBLSO, NPSO, and CSO for 20, 20, 20, 19, 14, 14, and 12 examples and is defeated by them only for 0,0,0,1,6,6,8 examples, respectively. Summarily, DVENN significantly outperforms PSO, ACO, GPBO, and JOA, while winning RBLSO, NPSO, and CSO. It, as a new-type optimization approach, not only has a stronger optimization performance than each compared approach, but also can comprehensively adapt to multiple kinds of complicated LSGO problems.

Related to those theoretical optimal objective values in Table 1, the curves in Figure 2 clearly validate that the examples in the CEC'2013 suite are extremely difficult in solving their optima. The characteristics of the objective functions' inseparability and the variables' rotation and overlapping seriously restrict the approaches' abilities to approach the optima in the later stage of population evolution. Even so, the DVENN's optimization performance is stronger than that of each compared approach, and thus it can adapt to multiple kinds of LSGO problems. The compared approaches are easily trapped into local minima, which clearly validates the fact that

Table 3. The CEC'2013 suite: Comparison of statistical results acquired by the eight approaches

<i>f</i>	<i>Q</i>	RBLSO	PSO	ACO	GPBO	JOA	NPSO	CSO	DVENN
<i>f</i> ₁	μ	1.15E-20↑	3.06E+11↑	3.31E+11↑	5.51E+00↓	9.51E-10↓	4.31E-21↓	4.06E-17↓	1.07E+03
	σ	5.54E-22	1.58E+10	6.82E+10	1.08E+01	2.12E-10	5.56E-22	1.94E-18	8.07E+02
<i>f</i> ₂	μ	7.99E+02↑	1.08E+05↑	8.96E+04↑	4.60E+02↑	2.46E+04↑	3.39E+03↑	9.46E+02↑	0.00E+00
	σ	3.09E+01	3.01E+03	2.97E+03	2.65E+01	2.55E+03	1.54E+02	4.92E+01	0.00E+00
<i>f</i> ₃	μ	2.16E+01↑	2.15E+01↑	2.16E+01↑	2.00E+01↑	2.12E+01↑	2.16E+01↑	2.16E+01↑	1.21E-03
	σ	5.60E-03	1.91E-02	1.01E-02	5.73E-04	3.64E-01	2.70E-03	6.40E-03	3.4E-04
<i>f</i> ₄	μ	9.66E+09↑	4.94E+12↑	1.95E+13↑	8.32E+12↑	2.82E+10↑	5.26E+09↑	1.18E+10↑	0.00E+00
	σ	1.59E+09	2.16E+12	5.17E+12	2.59E+12	7.96E+09	4.95E+08	1.42E+09	0.00E+00
<i>f</i> ₅	μ	7.05E+05↑	3.25E+07↑	6.29E+07↑	3.06E+07↑	3.75E+07↑	1.23E+06↑	8.07E+05↑	0.00E+00
	σ	1.25E+05	3.56E+06	3.98E+06	4.87E+06	5.06E+06	2.84E+05	1.12E+05	0.00E+00
<i>f</i> ₆	μ	1.06E+06↑	1.06E+06↑	1.07E+06↑	1.06E+06↑	1.05E+06↑	1.06E+06↑	1.06E+06↑	2.49E-03
	σ	1.04E+03	8.78E+02	2.02E+03	1.13E+03	5.84E+03	1.19E+03	1.20E+03	1.23E-04
<i>f</i> ₇	μ	7.88E+06↑	5.16E+12↑	3.13E+15↑	1.97E+10↑	2.57E+08↑	8.07E+06↑	1.31E+07↑	0.00E+00
	σ	3.41E+06	4.56E+12	1.94E+15	8.33E+09	7.78E+07	2.79E+06	5.35E+06	0.00E+00
<i>f</i> ₈	μ	1.59E+14↑	9.64E+16↑	1.04E+18↑	3.66E+20↑	5.67E+14↑	9.22E+13↑	3.09E+14↑	0.00E+00
	σ	2.67E+13	4.21E+16	3.58E+17	1.83E+21	1.75E+14	2.16E+13	7.22E+13	0.00E+00
<i>f</i> ₉	μ	1.22E+08↑	4.31E+15↑	1.53E+16↑	5.05E+09↑	1.32E+10↑	2.10E+08↑	1.39E+08↑	0.00E+00
	σ	3.50E+07	3.52E+15	7.79E+15	9.87E+08	1.53E+09	4.75E+07	3.17E+07	0.00E+00
<i>f</i> ₁₀	μ	9.40E+07↑	9.43E+07↑	9.60E+07↑	9.40E+07↑	9.25E+07↑	9.43E+07↑	9.08E+07↑	1.51E-04
	σ	2.62E+05	2.53E+05	3.19E+05	2.03E+05	7.00E+05	2.86E+05	9.09E+05	2.34E-05
<i>f</i> ₁₁	μ	9.29E+11↑	2.54E+14↑	7.75E+15↑	1.13E+12↑	1.47E+09↑	9.26E+11↑	9.35E+11↑	0.00E+00
	σ	1.78E+10	1.05E+14	2.34E+15	5.84E+11	3.79E+08	9.35E+09	9.81E+09	0.00E+00
<i>f</i> ₁₂	μ	1.16E+03↑	7.98E+12↑	8.00E+12↑	1.88E+04↑	3.60E+03↑	1.69E+03↑	1.87E+03↑	1.03E+03
	σ	5.85E+01	2.12E+11	1.78E+11	4.32E+03	1.26E+03	1.13E+02	1.44E+02	1.47E-02
<i>f</i> ₁₃	μ	1.21E+09↑	1.23E+15↑	2.73E+17↑	8.90E+10↑	6.37E+09↑	4.32E+08↑	7.59E+08↑	0.00E+00
	σ	4.66E+08	3.34E+14	1.07E+17	2.54E+10	1.25E+09	1.81E+08	2.37E+08	0.00E+00
<i>f</i> ₁₄	μ	5.42E+09↑	1.64E+16↑	4.64E+17↑	2.24E+12↑	2.41E+10↑	6.99E+07↑	5.85E+09↑	0.00E+00
	σ	2.53E+09	5.80E+16	2.13E+17	5.37E+11	1.15E+10	7.42E+06	3.62E+09	0.00E+00
<i>f</i> ₁₅	μ	3.43E+07↑	1.23E+17↑	1.24E+17↑	2.57E+10↑	3.21E+09↑	3.39E+06↑	5.24E+07↑	0.00E+00
	σ	4.12E+06	8.63E+16	4.99E+16	2.06E+10	7.32E+09	2.82E+05	3.68E+06	0.00E+00
w/t/l		13/1/1	15/0/0	15/0/0	15/0/0	13/1/1	13/1/1	13/1/1	–

The bolded values represent the optimal results in the cec2013 standard test suite.

metaheuristics easily incur the phenomenon of premature convergence when faced with LSGO problems. Relatively speaking, NPSO, RBLSO, and CSO are more competitive by comparison with GPBO, JOA, PSO, and ACO. On the other hand, DVENN can make full use of the output activities of four subnetworks to regulate the states' transition amplitude and direction, while exploiting potential states multi-directionally. Thereby, it, with a high convergence rate, can avoid the phenomenon of premature convergence as much as possible.

Case study: Operational amplifier optimization design

The experimental analyses in the previous section clearly show that DVENN is a competitive optimizer for LSGO problems. However, it keeps open whether the approach can effectively handle complex engineering design optimization problems. Hereby, the design of an operational amplifier in the analog circuited circuit is modeled and solved by DVENN and the compared approaches.

Architecture on operational amplifier

The operational amplifier (op-amp), say a high-gain differential amplifier, is an indispensable part of any analog or mixed-signal system. Various kinds of complex operational amplifiers can be used to achieve a great number of functions, e.g., signal amplification, filtering, power generation, analog-to-digital conversion, etc. Especially, since two-stage operational amplifiers are simple,

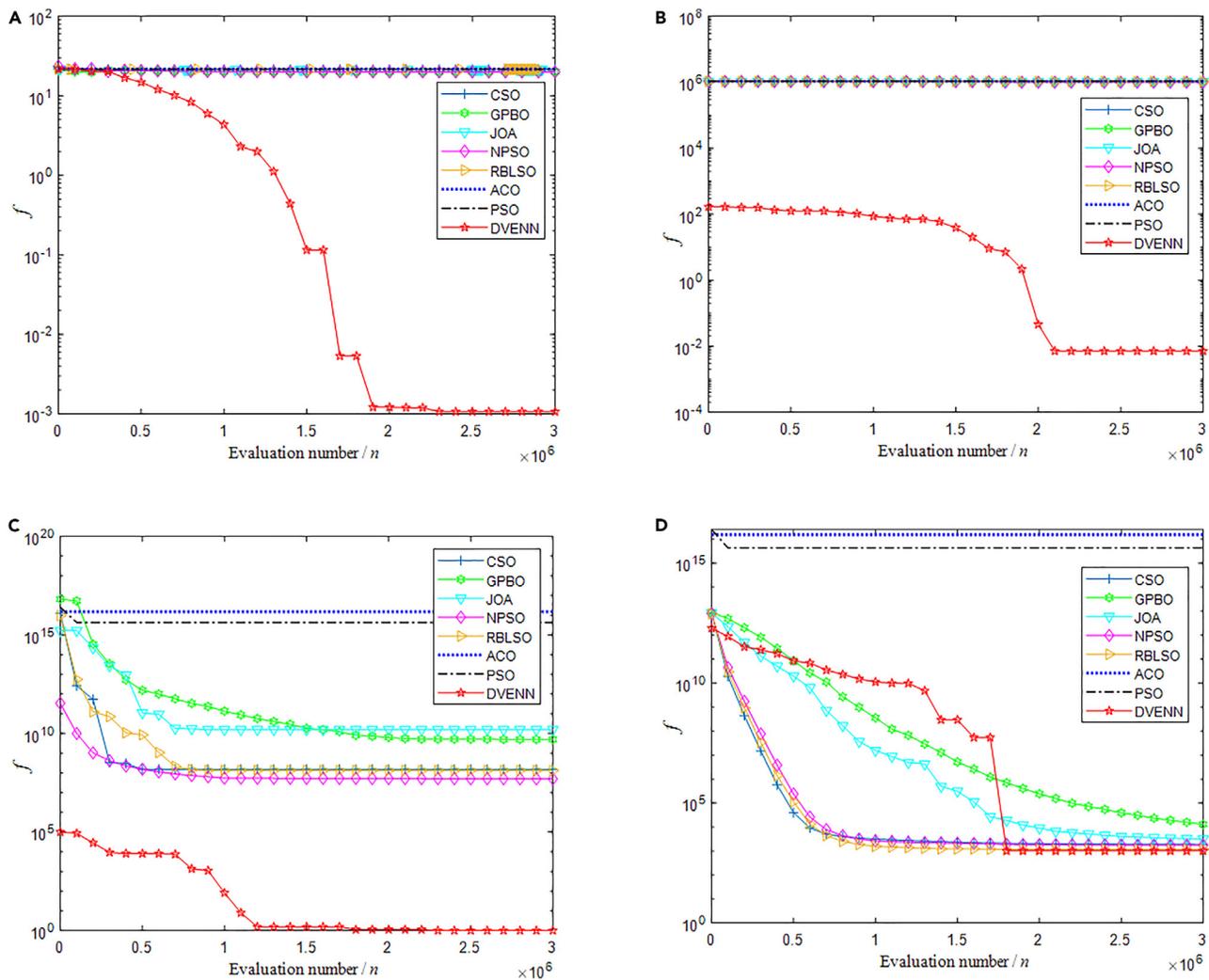


Figure 2. Comparison of the approaches' average search curves for the four examples in the CEC'2013 suite

- (A) f_3 .
- (B) f_6 .
- (C) f_9 .
- (D) f_{12} .

popular, and practical in engineering applications, they have been a common op-amp whose circuit structure is usually given in Figure 3.

Based on the design requirements and Figure 3, the design process of an op-amp can be divided into two main steps. One is to construct or select an appropriate basic circuit structure and determine the related structure diagram of transistors interconnect, and the other is to define all required parameter settings rationally, e.g., DC bias current, transistor size, compensation circuit, etc. In the design-oriented requirements, the main concern concentrates on the performance indices of the designed op-amp, e.g., gain (A_v), unity gain bandwidth (UGB), setup time, swing rate (SR), input common-mode circumference ($ICMR$), common mode rejection ratio ($CMRR$), power supply rejection ratio ($PSRR$), output voltage swing, output resistance, offset, noise, layout area, etc.

Architecture and requirement on the open-loop op-amp

In the standard two-stage CMOS op-amp as in Figure 4, all transistors, which consist of $V \rightarrow I$ and $I \rightarrow V$ cascades without output buffering, are assumed to operate in the saturation region. The first gain stage consists of two differential amplifiers with the same transistors, which converts differential mode voltages into currents to be outputted through the current mirror load. The first stage's circuit incorporates two n-type MOSFETs, M1 and M2, with the same sizes, but the current mirror involves in two p-type MOSFETs, M3 and M4, with the same sizes.

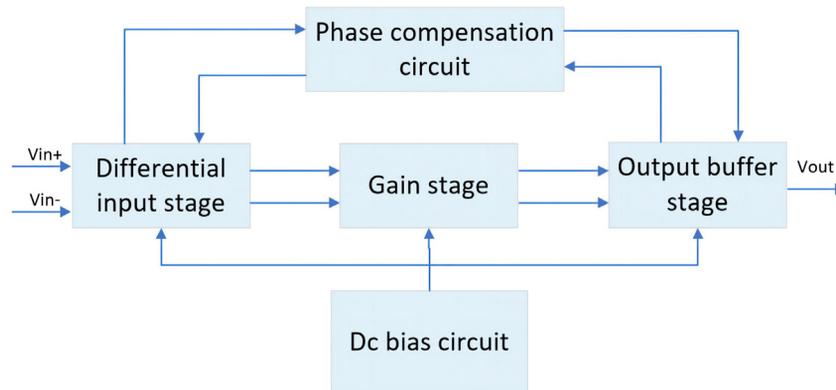


Figure 3. Block diagram on a two-stage operational amplifier

Related to Figure 4, the M1 and M2's gate-source voltages at the differential input endpoints, i.e., VGS1 and VGS2, are the same under static conditions, while the sum of their currents is equal to the current of the M5 tube. The differential pairs M1 and M2 as well as the current mirror loads M3 and M4 complete the conversion of $V \rightarrow I$ and $I \rightarrow V$ electrical signals, respectively. The current mirror with a large output resistor improves the gain of the first stage and converts the differential mode output signal of the differential amplifier into a single-ended signal. The design performance parameters need to satisfy the specific requirements as in Table 4.

Optimization model design on the two-stage op-amp

Based on the design principle and analysis of the analog circuit, the detailed two-stage open-loop op-amp model is given below.

Low-frequency gain index. The low-frequency gain is computed by

$$LFG(x) = 20 \lg \left(\frac{2g_{m1}g_{m6}}{x_9 I_6 (\lambda_2 + \lambda_4) (\lambda_6 + \lambda_7)} \right), \tag{Equation 3}$$

with $x = (x_1, \dots, x_{10}) = (S_1, \dots, S_8, I_5, C_C)$, where S_k denotes the proportion of width and length of the k -th MOSFET transistor with $1 \leq k \leq 8$; I_5 and C_C stand for the attenuation current of the fifth transistor and the bias current, respectively. Besides, I_6 is the output current of the sixth transistor, given by

$$I_6 = g_{m6}^2 / 2K_p' x_3. \tag{Equation 4}$$

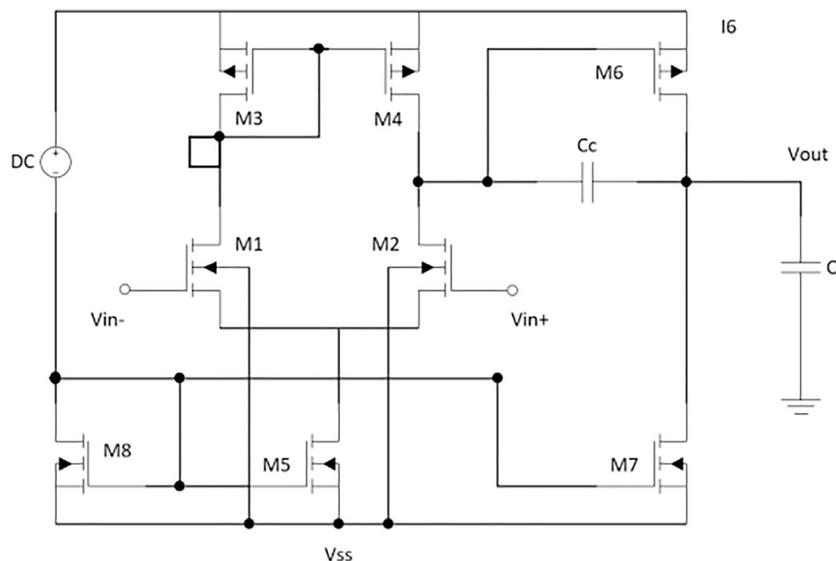


Figure 4. Circuit diagram of the two-stage op-amp

Table 4. Two-stage operational amplifier design characteristic requirements

No	Performance parameter	Requirement
1	area (TA)	<300um ²
2	gain (Av)	>50dB
3	Unity gain bandwidth (UGB)	>5MHz
4	Positive source suppression (PSRR+)	>60dB
5	Negative power supply rejection (PSRR-)	>60dB
6	The pendulum rate (SR)	>10V/us
7	Static power (P _s)	<2mW
8	The phase margin (PM)	>45°

λ_k represents the k -th transistor' design parameter with $k = 2, 4, 6$ and 7 ; g_{m1} and g_{m6} are denoted orderly by the transconductances of the first and sixth MOSFET transistors in Figure 4, namely

$$g_{m1} = (K'_n x_1 x_9)^{\frac{1}{2}}, \quad g_{m6} = \frac{g_{m4} x_6}{x_4}, \quad (\text{Equation 5})$$

with the NMOS and PMOS' intrinsic conduction factors K'_n and K'_p , where g_{m4} is the transconductance of the fourth MOSFET transistor, given by

$$g_{m4} = (K'_p x_3 x_9)^{\frac{1}{2}}. \quad (\text{Equation 6})$$

Again, since the low-frequency gain is required to be beyond 50 dB, the gain performance index is expressed by

$$f(x) = \max\left(20 \lg\left(\frac{2g_{m1}g_{m6}}{x_9 I_6 (\lambda_2 + \lambda_4) (\lambda_6 + \lambda_7)}\right), 50\right). \quad (\text{Equation 7})$$

Constraint condition

(1) Gain bandwidth constraint. The unit gain bandwidth is computed by

$$UGB = \frac{g_{m1}g_{m6}}{x_{10} + A_2 C_{gd6}}, \quad (\text{Equation 8})$$

where A_2 and C_{gd6} orderly refer to the second stage gain and the gate-drain capacitance of the sixth transistor M₆, defined by

$$A_2 = \frac{g_{m6}}{I_6 (\lambda_6 + \lambda_7)}, \quad C_{gd6} = CGD \times L \times x_6, \quad (\text{Equation 9})$$

with the transistor channel length L . Based on the design requirements in Table 4, $EUGB$ is required to be larger than 5MHz, and thus the design of the two-stage op-amp needs to satisfy the following constraint:

$$g_1(x) = \frac{5}{UGB} = \frac{5(x_{10} + A_2 C_{gd6})}{g_{m1}g_{m6}} < 1. \quad (\text{Equation 10})$$

(2) Slew rate constraint. The slew rate as the speed index of the op-amp denotes the conversion rate of the output voltage. Since $SR > 10V/us$, the below constraint needs to be satisfied, given by

$$g_2(x) = \frac{10}{\frac{x_9}{x_{10}}} < 1. \quad (\text{Equation 11})$$

(3) Layout area constraint. The layout area of the op-amp, TA , is decided by

$$TA = L^2 \sum_{i=1}^8 x_i. \quad (\text{Equation 12})$$

Since $TA < 300\mu\text{m}^2$, the below constraint holds:

$$g_3(\mathbf{x}) = \frac{TA}{300} < 1. \quad (\text{Equation 13})$$

(4) The power supply rejection ratio constraint. $PSRR^+$ and $PSRR^-$ are orderly the positive and negative ratios of the changes in the input power supply, computed by.

$$PSRR^+ = 2g_{m1}g_{m6} / x_9 I_6 (\lambda_2 + \lambda_4) \lambda_6, \quad (\text{Equation 14})$$

and

$$PSRR^- = 2g_{m1}g_{m6} / x_9 I_6 (\lambda_2 + \lambda_4) \lambda_7. \quad (\text{Equation 15})$$

Since $PSRR^+$ and $PSRR^-$ are larger than 60dB, there exists two constrains below:

$$g_4(\mathbf{x}) = \frac{60}{20 \lg\left(\frac{2g_{m1}g_{m6}}{x_9 \lambda_6 I_6 (\lambda_2 + \lambda_4)}\right)} < 1, \quad (\text{Equation 16})$$

and

$$g_5(\mathbf{x}) = \frac{60}{20 \lg\left(\frac{2g_{m1}g_{m6}}{x_9 \lambda_7 I_6 (\lambda_2 + \lambda_4)}\right)} < 1, \quad (\text{Equation 17})$$

(5) Static power consumption constraint. P_s refers to the total power consumption caused by the op-amp, which, together with $P_s < 2\text{mW}$, satisfies

$$g_6(\mathbf{x}) = \frac{P_s}{2} = \frac{(VDD - VSS)(2x_9 + I_6)}{2000} < 1, \quad (\text{Equation 18})$$

with the positive power supply VDD and the negative power supply VSS .

(6) Phase margin constraint. The transfer function of the open-loop operational amplifier is given by

$$G(s) = \frac{a\left(1 - s\left(\frac{C_c}{g_{mII}}\right) - R_Z R_I\right)}{1 + bs + cs^2 + ds^3}, \quad (\text{Equation 19})$$

with the first stage resistance R_I , the second stage trans-conductance g_{mII} , and the zero resistance R_Z , where a , b , c , and d are the coefficients given by

$$a = g_{mI}g_{mII}R_I R_{II}, b = R_I(C_I + C_c) + R_{II}(C_{II} + C_c) + g_{mII}R_I R_{II} + R_Z C_c;$$

$$c = R_I R_{II}(C_I C_{II} + C_I C_c + C_{II} C_c) + R_Z C_c(R_I C_I + R_{II} C_{II}), d = R_I R_{II} R_Z C_I C_c.$$

with the second stage resistance R_{II} , the first and second capacitance $s C_I$ and C_{II} , and the first trans-conductance g_{mI} . Hence, the phase margin Φ_M of the operational amplifier is usually computed by

$$\Phi_M = \pi - \sum_{1 \leq i \leq 3} \tan^{-1}\left(\frac{UGB}{P_i}\right) - \tan^{-1}\left(\frac{UGB}{Z_0}\right). \quad (\text{Equation 20})$$

where Z_0 denotes the i -th zero point of the above transfer function. On the other hand, to ensure that $\Phi_M > 45^\circ$, we take $Z_0 = 10UGB$, while three constraints need to be satisfied:

$$g_7(\mathbf{x}) = \frac{10g_{m2}}{g_{m6}} < 1, \quad (\text{Equation 21})$$

Table 5. Comparison of statistical results acquired by the approaches

Algorithm	Best	Worst	Mean	Std
RBLSO	0.29995	0.31101	0.29997	0.01412
PSO	0.30002	0.32001	0.30103	0.01898
ACO	0.30004	0.30322	0.30012	0.00910
GPBO	0.30553	0.33689	0.31887	0.02242
JOA	0.29958	0.31895	0.30175	0.01734
NPSO	0.30561	0.32982	0.31102	0.01956
CSO	–	–	–	–
DVENN	0.27986	0.29856	0.28012	0.00844

The bolded values represent the best results among the test results of the optimized model of the two-stage operational amplifier.

$$g_8(x) = \frac{1.22C_L \left(\frac{g_{m2}}{x_{10}} \right)}{g_{m6}x_{10}} < 1, \tag{Equation 22}$$

$$g_9(x) = \frac{40UGBC_{gs3}}{g_{m4}} < 1, \tag{Equation 23}$$

where C_{gs3} is the coefficient of the gate-source capacitance. Besides, the fifth MOSFET transistor M5 provides the bias current for the first stage amplification circuit, while so does the seventh MOSFET transistor M7 for the second stage amplification circuit. To ensure the operational stability of the op-amp, the constraint below is needed to stabilize the working state of the circuit, namely

$$h(x) = \frac{x_7}{x_5} - \frac{l_6}{x_9} = 0. \tag{Equation 24}$$

Summarily, the two-state operational amplifier optimization model (TSOAOM) can be expressed by

$$\underset{x}{\text{Minimize}} f(x)$$

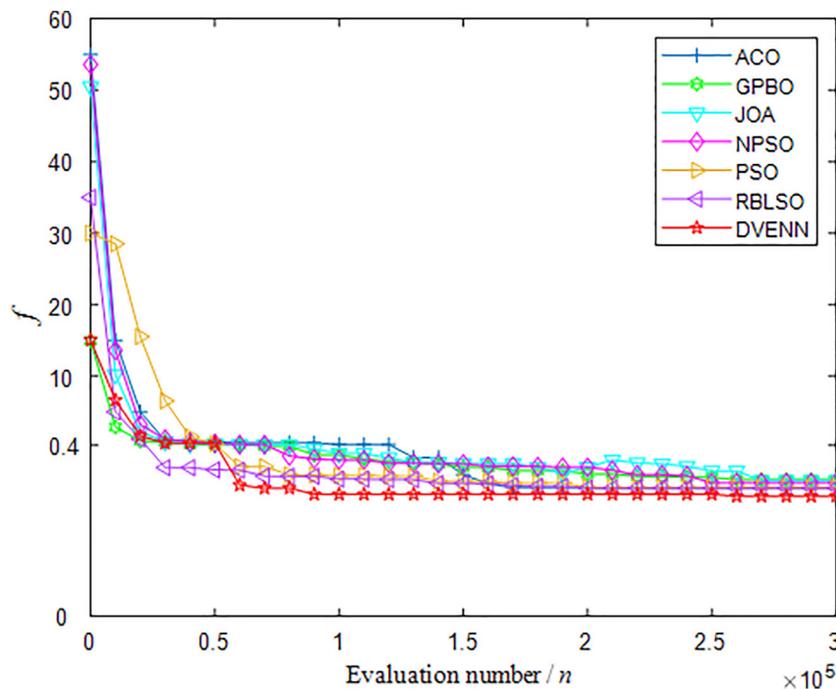


Figure 5. Comparison of average search curves

Table 6. Comparison of statistical results acquired by the eight approaches

Algori.	Area (TA)	A_v	UGB	PSRR+	PSRR-	SR	P_s	PM
RBLSO	292	92.82	9.39	98.28	91.37	14.31	1.32	49.5
PSO	290	90.57	8.84	89.8	87.44	15.82	1.32	49.9
ACO	295	90.65	9.13	96.3	89.81	16.05	1.34	47.9
GPBO	293	90.53	8.98	87.53	85.72	13.75	1.22	49.7
JOA	285	91.11	7.99	93.85	90.34	14.86	1.51	50.7
NPSO	288	87.14	9.25	94.8	89.65	15.58	1.13	51.6
CSO	–	–	–	–	–	–	–	–
DVENN	289	98.4	10.34	97.08	99.34	17.8	1.2	53.59

$$\text{s.t.} \begin{cases} g_i(x) < 1, 1 \leq i \leq 9, \\ h(x) = 0, x_k \in [1, 50], 1 \leq k \leq 8 \\ x_9 \in [0, 30], x_{10} \in [0, 10]. \end{cases}$$

One such programming model, which includes ten variables and ten nonlinear equality or inequality constraints, is a strongly nonlinear constrained optimization problem. The variables take their values in the above-mentioned intervals, due to the practical requirements.

Optimal design scheme for the two-stage op-amp

The above model includes four known parameters, i.e., transistor length L , load capacitance C_L , positive and negative operating voltages V_{DD} and V_{SS} , in which they take $2\mu\text{m}$, 10pF , 2.5V , and -2.5V , respectively. Like the above experiment in the section of 'numerical experiment', take the seven metaheuristic approaches mentioned in the previous section to participate in comparison with DVENN when tackling the above optimization model. Herein, the penalty function method is adopted to handle the nine constraints in TSOAOM with the penalty factor 10^3 .

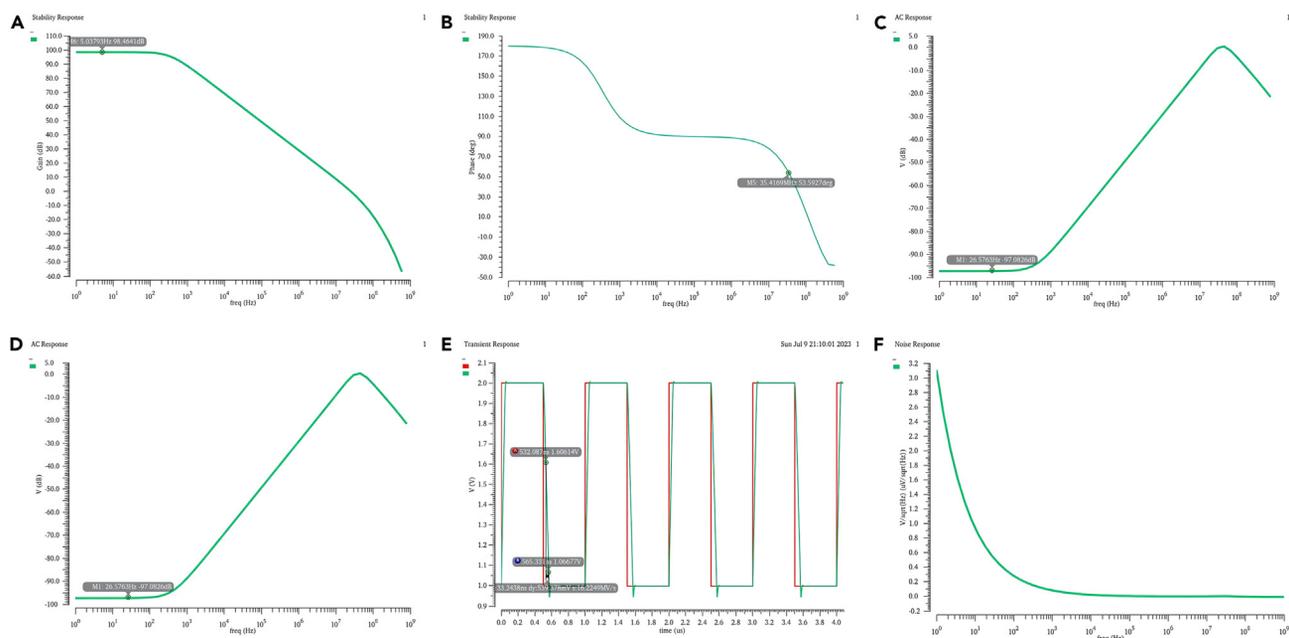


Figure 6. Characteristic diagram of each performance index of two-stage operational amplifier

- (A) Gain bandwidth.
- (B) Margin of phase.
- (C) PSRR⁺.
- (D) PSRR⁻.
- (E) SR.
- (F) Noise analysis.

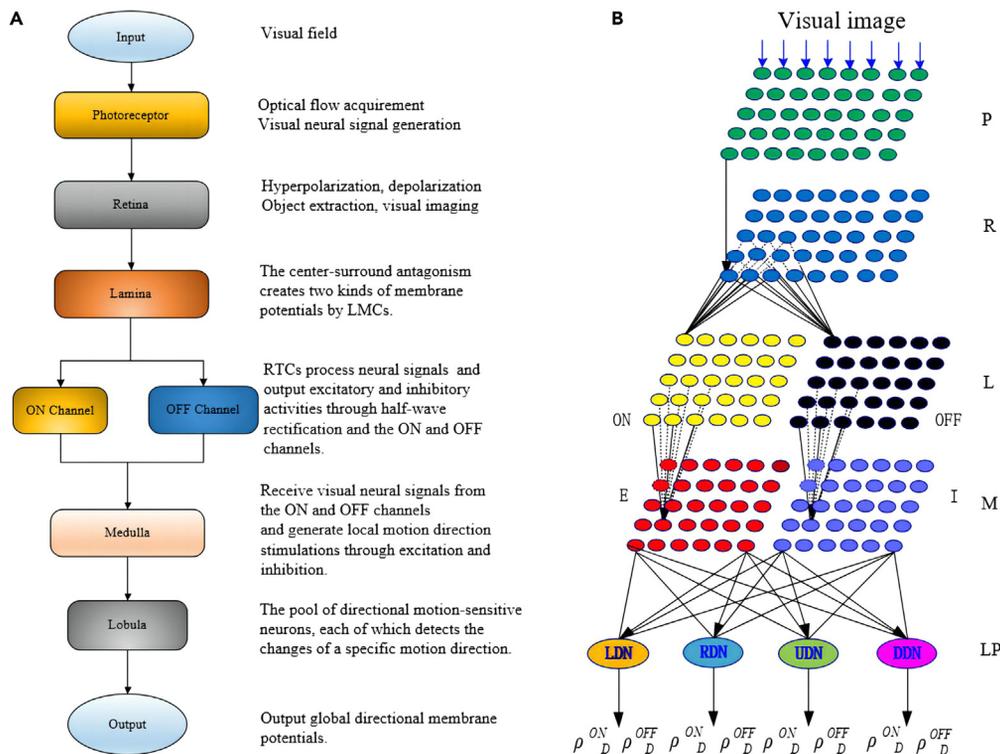


Figure 7. Schematic diagrams on visual response and the visual neural network

(A) Simplified dragonfly visual information processing.

(B) Dragonfly visual neural network.

Each approach with the total fitness evaluations 3×10^5 copes with TSOAOM twenty-five times independently, after which the acquired statistical results and convergence curves are given in Table 5 and Figure 5, respectively. It is highlighted that CSO is incomparable with the other seven approaches in that it cannot acquire any feasible solution or feasible op-amp design scheme per run. Nevertheless, the other seven approaches can all find feasible op-amp design schemes.

The results in Table 5 clearly emerge that DVENN can stably perform solution search with a high solution accuracy. On the one hand, its statistical results indicate that it can distinctively outperform the other six approaches with the aspect of solution search performance and thereby can acquire a high-quality op-amp design scheme after each execution, on the other hand it has the strong ability to excavate feasible solutions and the global optimal solution. The values in the bottom as in Table 5 indicate that it can acquire the best op-amp design scheme and perform a stable solution search. Besides, the compared approaches only have minor performance differences when seeking the optimal solution or the best design scheme. Relatively speaking, ACO and RBLSO are superior to PSO, GPBO, JOA, and NPSO, due to their solution search stability and solution quality.

The average search curves in Figure 5, generated by the approaches in the process of solution search clearly validate that DVENN and the compared approaches can perform stable solution search with different convergence speeds. Relatively, DVENN can find the best solution at a high global convergence rate, while being capable of finding approximate solutions. The best op-amp scheme, acquired by it can satisfy the performance design requirement of the op-amp. Consequently, DVENN is an available optimizer and can acquire the best design scheme to maximize the low-frequency gain index as much as possible in comparison with the compared approaches.

Empyrean Aether simulation

Herein, the Empyrean Aether simulator is used to examine whether the op-amp with the best op-amp design scheme acquired by DVENN, i.e., $x = (4, 4, 5, 5, 1, 50, 6, 2, 30, 1.6)$, can be simulated by the semi-empirical model. The simulation results in Table 6 show that the op-amp outputs a gain of 98.4dB, a unity gain bandwidth of 10.34MHz, and a phase margin of 53.59°. Furthermore, its positive and negative power supply rejection ratios are 97.08dB and 99.34dB, and SR is 17.8 v/us. These comply with the design characteristic requirements in Table 4.

The results in Table 6 demonstrate that except CSO, all the other approaches can obtain proper design results that satisfy the parameters' design requirements. RBLSO performs better by comparison against any of the other compared approaches except DVENN. We notice a fact, namely the larger the op-amp's gain, the smaller the phase margin. This requires that the acquired design scheme can achieve the trade-off between gain and phase margin. Nevertheless, while generating the largest phase margin, NPSO makes the op-amp output the smallest gain value, which means that its design scheme cannot take a rational balance between gain and phase margin. Consequently, it behaves poorly by

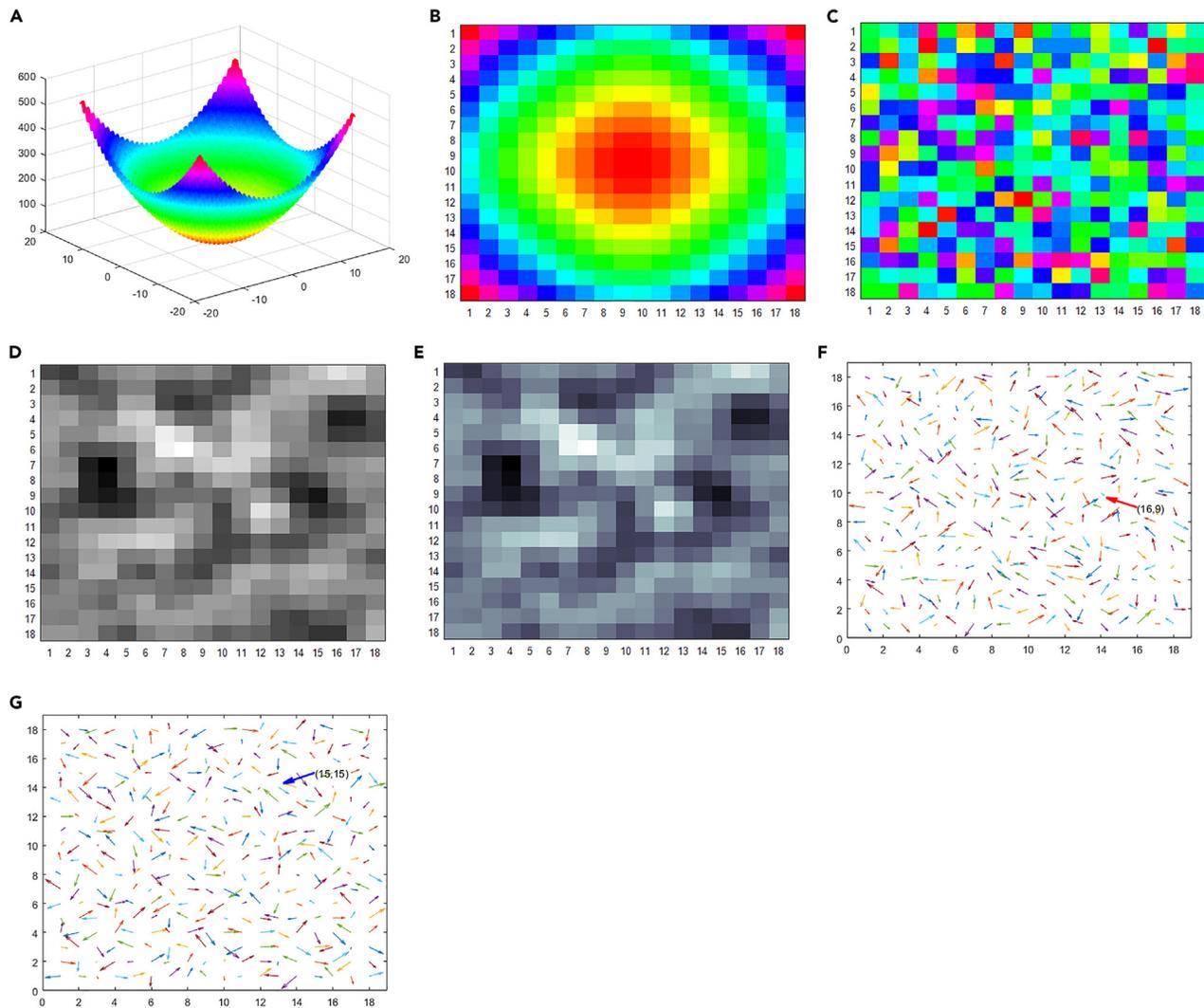


Figure 8. Schematic illustration on the L-DVNN's visual information processing

- (A) Function image.
- (B) P-layer.
- (C) R-layer.
- (D) L-layer/ON-channel.
- (E) L-layer/OFF-channel.
- (F) M-layer/ON local directions.
- (G) M-layer/OFF local directions. Global left ON motion direction. Global left OFF motion direction.

comparison against the other compared approaches. DVENN can balance the values of the phase margin and gain, and thus obtains a more appropriate performance result. Based on the results of SR, DVENN can make the op-amp operate at a proper speed, while the values of parameter design indices acquired can ensure such an op-amp to behave well. The power supply rejection ratio, as an important design index called the power supply rejection ratio (PSRR), can be used to evaluate the op-amp's ability to suppress the change of the supply voltage. The results in Table 6 hint that RBLSO, JOA, NPSO, and DVENN can all make the op-amp output proper suppression ratios.

Take the design scheme acquired by DVENN for example to plot the curves in Figure 6 below outputted by the op-amp in the simulation environment of the Empyrean Aether.

The stimulation curves validate that the design scheme can ensure that all the performance indices of the op-amp can coincide with the design requirements in Table 4, while the low-frequency gain maximization design can be achieved under the restrictions of the constraint performance indices. The numerical results illustrate that the design scheme acquired by DVENN is feasible and effective for the structure design of the op-amp in the analog integrated circuit. Therefore, DVENN is of potential value in simplifying the design of an integrated circuit and shortening the development cycle.

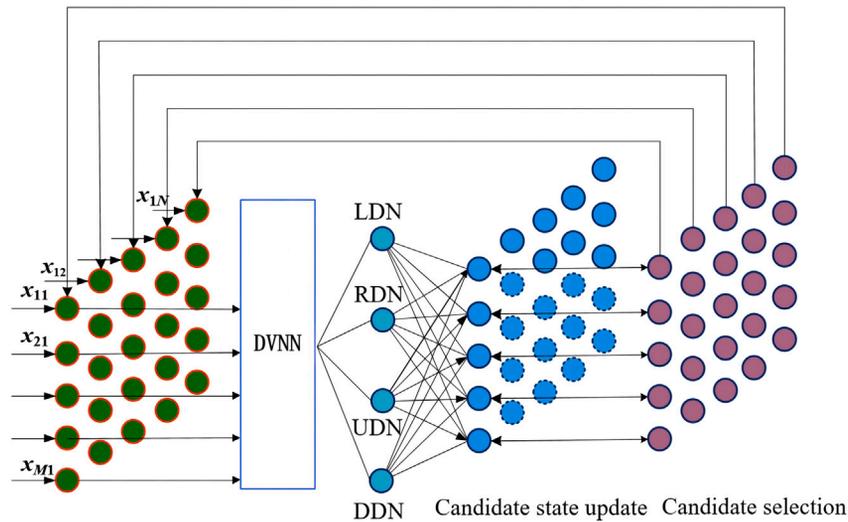


Figure 9. Schematic diagram on DVENN

RESULT AND DISCUSSION

Related to the above thirty-five examples with dimension 1000 or 2000 per one, the results on mean and variance acquired for each example in the section of 'numerical experiment' have sufficiently revealed the eight approaches' intrinsic performance characteristics, advantages, and disadvantages in tackling large-scale, inseparable, pathological, multi-modal function optimization problems. The t-test based statistical results show that there exist significant differences regarding the approaches' convergence as well as stability. DVENN can effectively solve almost all the thirty-five benchmark examples and can thus adapt to multiple kinds of complicated LSGO problems. Also, it can significantly outperform each compared approach for at least twenty-five examples and at most thirty-five ones. Precisely, it is superior to PSO, ACO, and GPBO absolutely while orderly outperforming RBLSO, JOA, NPSO, and CSO for 27, 32, 27 and 25 examples. Since the compared approaches cannot break through the shackle of local convergence, they usually get stuck into local optima when confronted with complicated LSGO problems.

The acquired results in the case study show that the eight approaches have different abilities for constraint handling and seeking the optimal solution. Precisely, the approaches except CSO can find feasible solutions in the process of solution search, while their solution qualities have no significant difference. Relatively, DVENN can acquire a higher-quality and more available design scheme for the design of the two-stage op-amp. Also, it generates a phase margin of 53.59° computed by (Equation 19), and thus the acquired design scheme is available. Besides, CSO cannot find any feasible solution, and thereby, it cannot acquire any feasible design scheme. The other compared approaches can obtain available and similar design schemes.

Based on the best solution acquired by DVENN, the op-amp is simulated under the Empyrean Aether simulation environment. The stimulation curves indicate that such an op-amp with a wide range of portability is effective and practical in engineering applications. This can not only greatly simplify the analog integrated circuit design in terms of the specified indices of gain, swing rate, etc., but also reduce the development cycle of op-amp design and the number of analog integrated circuit experiments under given requirements.

Conclusion

To cope with LSGO problems, the current work develops an interdisciplinary approach-dragonfly visual evolutionary neural network (DVENN), and apply it to the strongly nonlinear structure design of the op-amp in the analog integrated circuit. Firstly, inspired by the dragonfly visual response characteristics, a visual neural network (DVNN) is constructed to characterize behavioral changes in visual motion. It can not only measure changes in visual motion, but also online output four pairs of motion direction activities named learning rates. Secondly, a state matrix-based update strategy, which combines with DVNN to form DVENN, is skillfully designed to update the current state matrix. As an interdisciplinary optimizer with fast convergence, DVENN only includes the parameter of input resolution $M \times N$, while possessing some striking characteristics, e.g., rapid solution search, matrix-based state update, etc. Three conclusions can be drawn.

- (1) DVNN can online characterize the states' behavioral changes in terms of the feedback of state update information, by which the adjustable parameters as in DVENN can be updated adaptively. Herein, the four subnetworks output respective global learning rates to collaboratively guide the states' transition so that those potential states can be found as fast as possible. On the other hand, the state update scheme can promptly generate diverse states and discover high-quality solutions. DVENN, which integrates DVNN with the differential evolution-like state update scheme, can emerge the strong ability of global convergence and adapt to multiple kinds of LSGO problems. The theoretical analysis verifies that MOVENN is a low-computational-complexity approach with computational complexity $O(4MN(5D + m))$
- (2) Related to the two benchmark suites with thirty-five hard LSGO examples, the comparative experiments indicate that DVENN can effectively solve thirty-two of the thirty-five examples and converge to the global approximate or optimal solutions of twenty-seven

examples; RBLSO, PSO, ACO, GPBO, JOA, NPSO, and CSO can find the approximate solutions for 6, 0, 0, 0, 2, 2, and 6 examples, respectively. Thereby, we can conclude that DVENN is a strongly competitive optimizer while the compared approaches easily get stuck into local optima in the case where LSGO problems are solved. Relatively, RBLSO and CSO outperform PSO, ACO, GPBO, JOA, and NPSO. Summarily, the reason why DVENN can effectively solve LSGO problems experimentally include three points: (i) DVNN can online output four kinds of motion-directional activities to purposely guide each current state to generate four candidate states, which can promote the capability of excavating valuable candidate solutions; (ii) the state update scheme can significantly enhance the ability of global convergence; and (iii) DVENN is a competitive optimizer for LSGO problems.

- (3) The case study shows that, in comparison with the compared approaches, DVENN can acquire the best design scheme satisfying all the given design requirements, while the two-stage op-amp can generate a phase angle of 53.59° to achieve the desired design objective.

Finally, whereas DVENN as a new-type and competitive optimizer can comprehensively adapt to multiple kinds of MOPs, it also exposes some disadvantages. For example, the stability of search effect and the ability of global optimization need to be further improved for special LSGO problems with the characteristics of variable rotation and inseparability. On the other hand, the experimental results have sufficiently validated that, even though DVENN can solve LSGO problems with dimension 2000, the difficulty of solving ultrahigh dimensional LSGO problems need to be solved so that it can well adapt to optimizing the structure of deep learning networks.

Limitations of the study

Although the algorithm obtained in this paper is of strong application, its stability on solution search still needs to be enhanced due to the structural design of its dragonfly visual neural network. Some improvements need to be further done, to enable DVENN to tackle with such ultra-high-dimensional LSGO problems as the CNN's parameter optimization.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- [KEY RESOURCES TABLE](#)
- [RESOURCE AVAILABILITY](#)
 - Lead contact
 - Materials availability
 - Data and code availability
- [EXPERIMENTAL MODEL AND SUBJECT DETAILS](#)
- [METHOD DETAILS](#)
 - Dragonfly visual neurophysiology
 - Architecture on DVNN
 - L-DVNN
 - Other neural subnetworks
 - Dragonfly visual evolutionary neural network
 - State update
 - Algorithm formulation on DVENN
- [QUANTIFICATION AND STATISTICAL ANALYSIS](#)

ACKNOWLEDGMENTS

The authors would like to thank the cordial editors and reviewers for their contributions to this work. The current work is supported by National Natural Science Foundation of China (62063002) and Guizhou Science and Technology Projects (Grant numbers Qiankehe Support [2023] General 251).

AUTHOR CONTRIBUTIONS

H.W. performed the study and drafted the manuscript. Z.H.Z. as the corresponding author conceptualized the study and performed writing correction as well as funding acquisition.

DECLARATION OF INTERESTS

The authors declare that they have no known competing financial interests to influence the current work.

Received: August 30, 2023

Revised: December 5, 2023

Accepted: January 23, 2024

Published: January 29, 2024

REFERENCES

1. Jia, J., Yuan, S., Shi, Y., Wen, J., Pang, X., and Zeng, J. (2022). Improved sparrow search algorithm optimization deep extreme learning machine for lithium-ion battery state-of-health prediction. *iScience* 25, 103988.
2. Wang, X.L., Wang, Y.P., Cai, K., and Lai, J.-F. (2017). Off-line time aware divisible-load scheduling optimization model. *J. Univ. Electron. Sci. Technol. China* 46, 88–95.
3. Lan, B.X., and Wang, T.S. (2016). Optimization model and algorithms for large-scale rail passenger transport network operation. *Chinese Journal of Management Science* 24, 159–170.
4. Zhang, Y., Xie, J., Liu, Y., Li, C., Xiao, J., Ma, H., and Zhou, J. (2022). An immune chaotic adaptive evolutionary algorithm for energy-efficient clustering management in LPWSN. *Journal of King Saud University-Computer and Information Sciences* 34, 8297–8306.
5. Lagrange, A., Fauvel, M., and Grizonnet, M. (2017). Large-scale feature selection with Gaussian mixture models for the classification of high dimensional remote sensing images. *IEEE Trans. Comput. Imaging* 3, 230–242.
6. Li, D., Liu, M., Gao, S., Shi, Y., Zhang, Y., Li, Z., Chiang, P.Y., Maloberti, F., and Geng, L. (2019). Low-noise broadband CMOS TIA based on multi-stage stagger-tuned amplifier for high-speed high-sensitivity optical communication. *IEEE Trans. Circuits Syst. I.* 66, 3676–3689.
7. Aljarah, I., Faris, H., and Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput.* 22, 1–15.
8. Zhang, Y.D., Satapathy, S.C., Zhang, X., and Wang, S.H. (2021). COVID-19 diagnosis via DenseNet and optimization of transfer learning setting. *Cognitive computation* 13, 1–17.
9. Wang, S.H., Satapathy, S.C., Zhou, Q., Zhang, X., and Zhang, Y.D. (2022). Community-acquired pneumonia recognition by wavelet entropy and cat swarm optimization. *J. Grid Comput.* 20, 1–18.
10. Sabir, Z., Said, S.B., Al-Mdallal, Q., and Ali, M.R. (2022). A neuro swarm procedure to solve the novel second order perturbed delay Lane-Emden model arising in astrophysics. *Sci. Rep.* 12, 22607.
11. Sabir, Z., Raja, M.A.Z., Mahmoud, S.R., Guirao, J.L.G., and Sánchez, J.M. (2022). A heuristic computing approach using sequential quadratic programming to solve the fifth kind of induction motor model. *Fractals* 30, 2240240.
12. Saremi, S., Mirjalili, S., and Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in engineering software* 105, 30–47.
13. Cao, Z., Wang, L., Hei, X., Jiang, Q., Lu, X., and Wang, X. (2016). A phase based optimization algorithm for big optimization problems. In 2016 IEEE congress on evolutionary computation (CEC) (IEEE), pp. 5209–5214.
14. Cheng, R., and Jin, Y. (2015). A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.* 45, 191–204.
15. Cao, Z., Wang, L., and Hei, X. (2018). A global-best guided phase based optimization algorithm for scalable optimization problems and its application. *Journal of Computational Science* 25, 38–49.
16. Sun, G., Zhao, R., and Lan, Y. (2016). Joint operations algorithm for large-scale global optimization. *Appl. Soft Comput.* 38, 1025–1039.
17. Xiao, T.Y., and Zhang, Z.H. (2021). Particle swarm optimization algorithm for solving large-scale function optimization. *Computer Engineering and Design* 42, 1614–1622.
18. Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks* 4, 1942–1948.
19. Dorigo, M., Di Caro, G., and Gambardella, L.M. (1999). Ant algorithms for discrete optimization. *Artif. Life* 5, 137–172.
20. Deng, H., Peng, L., Zhang, H., Yang, B., and Chen, Z. (2019). Ranking-based biased learning swarm optimizer for large-scale optimization. *Inf. Sci.* 493, 120–137.
21. Tang, M., Qiao, L., Huang, Z., Liu, X., Peng, Y., and Liu, X. (2020). Accelerating SGD using flexible variance reduction on large-scale datasets. *Neural Comput. Appl.* 32, 8089–8100.
22. Liu, Z., Shi, X., He, L., Yu, D., Jin, H., Yu, C., Dai, H., Feng, Z., and Feng, Z. (2020). A parameter-level parallel optimization algorithm for large-scale spatio-temporal data mining. *Distrib. Parallel Databases* 38, 739–765.
23. Maleknia, M., and Shamsi, M. (2020). A gradient sampling method based on ideal direction for solving nonsmooth optimization problems. *J. Optim. Theory Appl.* 187, 181–204.
24. Xu, S., and Zhu, M. (2023). Efficient Gradient Approximation Method for Constrained Bilevel Optimization. Preprint at arXiv. 2302.01970. <https://doi.org/10.48550/arXiv.2302.01970>.
25. Potter, M.A., and De Jong, K.A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.* 8, 1–29.
26. Jia, Y.H., Mei, Y., and Zhang, M. (2022). Contribution-based cooperative coevolution for nonseparable large-scale problems with overlapping subcomponents. *IEEE Trans. Cybern.* 52, 4246–4259.
27. Kaboli, S.H.A., and Alqallaf, A.K. (2019). Solving non-convex economic load dispatch problem via artificial cooperative search algorithm. *Expert Syst. Appl.* 128, 14–27.
28. Sun, L., Lin, L., Li, H., and Gen, M. (2019). Cooperative co-evolution algorithm with an MRF-based decomposition strategy for stochastic flexible job shop scheduling. *Mathematics* 7, 318.
29. Mahdavi, S., Rahnamayan, S., and Shiri, M.E. (2018). Incremental cooperative coevolution for large-scale global optimization. *Soft Comput.* 22, 2045–2064.
30. Wang, B., Ren, L., Prado, J.D., He, W., Jin, H., Jiang, Q., and Wang, X. (2021). An adaptive mechanism with cooperative coevolution and covariance for differential evolution. *IEEE Access* 9, 99890–99904.
31. Yin, Y., Wang, L., and Zhang, L. (2022). A Multipopulation Dynamic Adaptive Coevolutionary Strategy for Large-Scale Complex Optimization Problems. *Sensors* 22, 1999.
32. Cuevas, E., Cienfuegos, M., Zaldivar, D., and Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst. Appl.* 40, 6374–6384.
33. Luo, J., Ni, X., and Hu, Y. (2017). A hybrid differential evolution algorithm with multiple search strategies for large-scale optimization. *J. S. China Univ. Technol.: Natural Science Edition* 45, 97–103.
34. Ma, Y., and Bai, Y. (2020). A multi-population differential evolution with best-random mutation strategy for large-scale global optimization. *Appl. Intell.* 50, 1510–1526.
35. Shadravan, S., Naji, H.R., and Bardsiri, V.K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng. Appl. Artif. Intell.* 80, 20–34.
36. Sabir, Z., Raja, M.A.Z., Baleanu, D., and Guirao, J.L.G. (2022). Neuro-swarm computational heuristic for solving a nonlinear second-order coupled Emden–Fowler model. *Soft Comput.* 26, 13693–13708.
37. Sabir, Z., Baleanu, D., Ali, M.R., and Sadat, R. (2022). A novel computing stochastic algorithm to solve the nonlinear singular periodic boundary value problems. *Int. J. Comput. Math.* 99, 2091–2104.
38. Fu, B., He, Y., Guo, Q., and Zhang, J. (2023). An improved competitive particle swarm optimization algorithm based on de-heterogeneous information. *Journal of King Saud University-Computer and Information Sciences* 35, 101470.
39. Geurten, B.R.H., Nordström, K., Sprayberry, J.D.H., Bolzon, D.M., and O'Carroll, D.C. (2007). Neural mechanisms underlying target detection in a dragonfly centrifugal neuron. *J. Exp. Biol.* 210, 3277–3284.
40. Wiederman, S.D., Shoemaker, P.A., and O'Carroll, D.C. (2008). A model for the detection of moving targets in visual clutter inspired by insect physiology. *PLoS One* 3, e2784.
41. Hassenstein, B., and Reichardt, W. (1956). Systemtheoretische analyse der zeit-reihenfolgen-und vorzeichenbewertung bei der bewegungsperzeption des räusselfäfers chlorophanus. *Z. Naturforsch. B Chem. Sci.* 11, 513–524.
42. Reichardt, W. (1987). Evaluation of optical motion information by movement detectors. *J. Comp. Physiol.* 161, 533–547.
43. Egelhaaf, M., and Borst, A. (1989). Transient and steady-state response properties of movement detectors. *JOSA A* 6, 116–127.
44. Bagheri, Z.M., Cazzolato, B.S., Grainger, S., O'Carroll, D.C., and Wiederman, S.D. (2017). An autonomous robot inspired by insect neurophysiology pursues moving features in natural environments. *J. Neural Eng.* 14, 046030.
45. Wang, H., Peng, J., and Yue, S. (2020). A directionally selective small target motion detecting visual neural network in cluttered backgrounds. *IEEE Trans. Cybern.* 50, 1541–1555.
46. Ögmen, H., and Gagné, S. (1990). Neural network architectures for motion perception and elementary motion detection in the fly visual system. *Neural Network.* 3, 487–505.
47. Missler, J.M., and Kamangar, F.A. (1995). A neural network for pursuit tracking inspired by the fly visual system. *Neural Network.* 8, 463–480.
48. Zhang, Z., Yue, S., and Zhang, G. (2015). Fly visual system inspired artificial neural network for collision detection. *Neurocomputing* 153, 221–234.
49. Fu, Q., Bellotto, N., Hu, C., and Yue, S. (2018). Performance of a visual fixation model in an autonomous micro robot inspired by drosophila physiology. In 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO) (IEEE), pp. 1802–1808.
50. Wang, H., Peng, J., and Yue, S. (2017). An improved lptc neural model for background motion direction estimation. In 2017 Joint IEEE International Conference on

- Development and Learning and Epigenetic Robotics (ICDL-EpiRob) (IEEE), pp. 47–52.
51. Rind, F.C., and Bramwell, D.I. (1996). Neural network based on the input organization of an identified neuron signaling impending collision. *J. Neurophysiol.* *75*, 967–985.
 52. Yue, S., and Rind, F.C. (2013). Postsynaptic organisations of directional selective visual neural networks for collision detection. *Neurocomputing* *103*, 50–62.
 53. Hu, B., Yue, S., and Zhang, Z. (2017). A rotational motion perception neural network based on asymmetric spatiotemporal visual information processing. *IEEE Trans. Neural Netw. Learn. Syst.* *28*, 2803–2821.
 54. Fu, Q., Hu, C., Peng, J., and Yue, S. (2018). Shaping the collision selectivity in a looming sensitive neuron model with parallel on and off pathways and spike frequency adaptation. *Neural Netw.* *106*, 127–143.
 55. Brest, J., Zamuda, A., Fister, I., and Maučec, M.S. (2010). Large scale global optimization using self-adaptive differential evolution algorithm. In *IEEE congress on evolutionary computation (IEEE)*, pp. 1–8.
 56. LaTorre, A., Muelas, S., and Peña, J.M. (2013). Large scale global optimization: Experimental results with MOS-based hybrid algorithms. In *2013 IEEE congress on evolutionary computation (IEEE)*, pp. 2742–2749.
 57. Gray, P.R., Hurst, P.J., Lewis, S.H., and Meyer, R.G. (2009). *Analysis and Design of Analog Integrated Circuits* (John Wiley & Sons).
 58. Sherk, T.E. (1978). Development of the compound eyes of dragonflies (Odonata). III. Adult compound eyes. *J. Exp. Zool.* *203*, 61–80.
 59. Fabian, J.M., el Jundi, B., Wiederman, S.D., and O'Carroll, D.C. (2020). The complex optic lobe of dragonflies. Preprint at bioRxiv. <https://doi.org/10.1101/2020.05.10.087437>.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Microsoft Visual Studio2019	Microsoft Visual Studio	https://visualstudio.microsoft.com/zh-hans/free-developer-offers/
Microsoft Visio 2021	Microsoft	https://www.microsoft.com/zh-cn/microsoft-365/visio/flowchart-software
Empyrean Aether	Empyrean Aether	https://www.emyrean.com.cn/products/eda/analog.html
CEC2010 benchmark	CEC2010 benchmark	https://doi.org/10.1109/CEC.2010.5585927
CEC2013 benchmark	CEC2013 benchmark	https://doi.org/10.1109/CEC.2013.6557901

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact Zhuhong Zhang (zhzhang@gzu.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- All data reported in this paper will be shared by the [lead contact](#) upon reasonable request.
- This paper does not report original code.
- Any additional information required to reanalyze the data reported in this paper is available from the [lead contact](#) upon request.
- The work cites the benchmark examples to examine the acquired visual evolutionary neural network. Such examples are available. Please see <https://doi.org/10.1109/CEC.2010.5585927> and <https://doi.org/10.1109/CEC.2013.6557901>.

EXPERIMENTAL MODEL AND SUBJECT DETAILS

Based on the internal structure of dragonfly visual nerves and the physiological characteristics of dragonfly-specific motion detection neurons, this article attempts to build an artificial dragonfly visual neural network model, and develop a four-layer visual neural network. By utilizing the unique information processing mechanism of the dragonfly visual nervous system and the inspiration of swarm evolution, we designed a visual evolutionary neural network optimization algorithm for high-dimensional optimization, especially for LSGO problems, enabling us to discuss and solve nonlinear LSGO problems from the perspective of visual evolutionary neural network optimization. At the same time, this visual evolutionary neural network is combined with evolutionary ideas to study computational intelligence issues from an interdisciplinary perspective.

METHOD DETAILS

To develop a multi-output visual neural network, the next section introduces some basic dragonfly visual neurophysiologic theories, by which a simplified dragonfly visual information-processing mechanism is expounded by an illustrative schematic diagram. DVNN given in the later section implicates its topological architecture, functional module designs, and algorithm formulation.

Dragonfly visual neurophysiology

A dragonfly, with many more than 30,000 compound eyes,⁵⁸ has an unprecedentedly complex lobule, consisting of at least four consecutive synaptic neural posts and two lobular plate-like structures. Especially, the dragonfly visual system with a specific structure, as a visual information processing system, is responsible for receiving optical flow signals and transmitting hierarchical signals through specific visual neural mechanisms. Such a system receives optical flow signals and performs hierarchical information processing through five visual neural layers like any other insect, i.e., photoreceptor (P), retina (R), lamina (L), medulla (M), and lobular plate (LP).⁵⁹

The P-layer as a visual input layer is formed of photoreceptors, each of which perceives a brightness intensity at each moment and then transforms it into an electrical neural signal to be delivered to the matched retina node in the retina layer.

The R-layer is composed of ommatidia, each of which collects the neural signal generated by the counterpart at each moment in the P-layer. It not only judges whether a moving object presents in a visual field, but also performs figure-background discrimination in order to extract valuable visual signals. Therefore, it possesses the properties of object extraction and noise elimination.

The L-layer generates two kinds of response activities through the ON and OFF channels after collecting the outputs of all the retina nodes in the R-layer. It comprises two sublayers, i.e., the large monopole cell layer called the LMC-layer and the rectifier transient cell layer named the RTC-layer. The LMC-layer is formed of large monopole cells (LMCs), which produces excitatory membrane potentials through the ON channel, depending on the center-surround increment mechanism. The rectifier transient cell (RTC) layer is made of rectifier transient cells. Like the LMC-layer, each RTC also collects the activities of retina nodes around the counterpart on the R-layer, but generates its membrane potential through the OFF channel and the center-surround attenuation mechanism.

The M-layer incorporates two sublayers, excitation (E) and inhibition (I) layers. The layer E directly collects the output activities from the LMC-layer, and then outputs the changes of excitatory membrane potentials by means of the current and time-delayed membrane potentials. However, so does the layer I from the RTC-layer, and then outputs the changes of inhibitory membrane potentials. The main role of the M-layer is to sort the signals from the ON and OFF channels, depending on the stimulation and inhibition mechanism.

The LP-layer consists of lobular plate tangential cells (LPTCs), especially the four horizontal and vertical direction motion detection neurons, i.e., left detection neuron (LDN), right detection neuron (RDN), upper detection neuron (UDN), and downward detection neuron (DDN). Such four neurons can jointly detect the global motion direction changes of one or multiple moving objects in the left, right, upper, and downward motion directions, respectively. When some moving object approaches the dragonfly from left to right, right to left, top to bottom, or bottom to top, one of the four neurons will respond quickly to the neural signals from the M-layer, and later, generates a global motion direction activity to characterize the movement behavior. Since the LP-layer are almost covered by the dendrites of LPTCs, a network of connections between LPTCs is formed to combine the LPTCs' response activities to reflect visual motion changes in the whole field of view.

Summarily, related to the above neurophysiological findings, the dragonfly visual response mechanism can be simply expounded by an illustrative schematic diagram in Figure 7A. This can generate multiple global motion direction activities to measure the status of motion direction change in the visual scene by virtue of DVNN below.

Architecture on DVNN

Related to the dragonfly's hierarchical information-processing mechanism, DVNN, schematically illustrated by Figure 7B consists of four motion direction detection neural subnetworks, namely left (L), right (R), upper (U) and down (D) neural subnetworks orderly called L-, R-, U-, and D-DVNNs. Herein, L- and R-DVNNs detect the changes of visual motion in the *left* and *right* directions as in the visual scene respectively, each of which generates a pair of global motion direction activities. L-DVNN with neuron LDN outputs a pair of ρ_L^{ON} and ρ_L^{OFF} at any moment. Also, so does R-DVNN with neuron RDN, and outputs a global right motion direction activity pair, i.e., ρ_R^{ON} and ρ_R^{OFF} . Similarly, U-, and D-DVNNs with respective neurons UDN and DDN are orderly employed to measure the changes of visual motion in the *upper* and *downward* directions, while outputting their respective motion direction activity pairs, i.e., $(\rho_U^{ON}, \rho_U^{OFF})$ and $(\rho_D^{ON}, \rho_D^{OFF})$. All the four neural subnetworks have the same topological structures and share the P-, R-, and L-layers. The main differences between them consist in that their lateral suppression mechanisms are different in the M-layer, since their neurons only respond to visual movements in their preferential motion directions. Take L-DVNN for example to expound their neural layer designs.

L-DVNN

Related to the neurophysiologic findings mentioned in the previous section, the L-DVNN's architecture is illustratively formulated by Figure 7B above. The design principle is summarized below.

P-layer

The P-layer as an input layer consists of $M \times N$ nodes arranged in a matrix. Node (i, j) with state/candidate solution $\mathbf{x}_{ij}^{(t)}$ as in D (i.e., the LSGO' decision region) receives the grayscale value $f(\mathbf{x}_{ij}^{(t)})$ at time t . Such $M \times N$ states constitute a state matrix $\mathbf{X}^{(t)}$, while all the corresponding grayscale values form a grayscale image $f(\mathbf{X}^{(t)})$. At the initial moment, i.e., $t = 1$, $\mathbf{x}_{ij}^{(t)}$ is randomly generated in D , and conversely, it denotes the state at the matched position (i, j) in the candidate selection sublayer of the state update layer as in DVNN.

R-layer

The layer, with the same structure as the above layer, is formed of $M \times N$ retina nodes. Each node collects the activity output of the counterpart in the P-layer, and detects the change of optical intensity. Precisely, after capturing the activity output $f(\mathbf{x}_{ij}^{(t)})$ at the node (i, j) in the P-layer, the retina node (i, j) outputs an excitatory or inhibitory activity by

$$R_{ij}(t) = f(\mathbf{x}_{ij}^{(t)}) - f(\mathbf{x}_{ij}^{(t-1)}), 1 \leq i \leq M, 1 \leq j \leq N, \quad (\text{Equation 25})$$

where the node is in an excitatory state if $R_{ij}(t) > 0$, and conversely, it keeps inhibitory.

L-layer

The layer extracts feature information from the R-layer, depending on the center-surround suppression mechanism and the intrinsic ON and OFF channels. Based on the neurophysiologic explanations, it includes the LMC- and RTC-layers, and outputs the ON and OFF activity matrices online.

- (1) In the LMC-layer, after receiving the output activity matrix from the R-layer, the center-surround suppression mechanism is used to generate two activity matrices by means of the mechanism of signal projection and two convolution kernels, i.e., excitatory convolution kernel C_{11} and inhibitory one C_{12} below:

$$C_{11} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad C_{12} = -C_{11}$$

The first matrix, say E_1 , is the excitatory activity matrix with size $M \times N$, in which the element (i, j) is acquired by C_{11} and the following excitatory feature extraction model,

$$L_{ij}^E(t) = \sum_{0 \leq k, l \leq 2} R(i+k, j+l, t) w_{kl}^E, \quad 1 \leq i \leq M, 1 \leq j \leq N, \quad (\text{Equation 26})$$

where w_{kl}^E denotes the weight at position (k, l) in C_{11} . The second one, say I_1 , is the inhibitory activity matrix with size $M \times N$, in which each element (i, j) is determined by (Equation 27) and C_{12} , namely

$$L_{ij}^I(t) = \sum_{0 \leq k, l \leq 2} R(i+k, j+l, t) w_{kl}^I, \quad 1 \leq i \leq M, 1 \leq j \leq N, \quad (\text{Equation 27})$$

where w_{kl}^I is the weight at position (k, l) in C_{12} .

- (2) The RTC-layer is composed of two independent and parallel channels, i.e., ON and OFF channels. Such two channels with $M \times N$ nodes per channel output excitatory and inhibitory activity matrices, respectively. The ON channel extracts the excitatory activities in the above excitatory activity matrix E_1 but inhibits those inhibitory activities by the mechanism of half-wave rectification. Precisely, its node (i, j) first receives the output of the counterpart in E_1 , i.e., $L_{ij}^E(t)$, and later, outputs an activity $L_{ij}^{ON}(t)$ by

$$L_{ij}^{ON}(t) = \begin{cases} L_{ij}^E(t), & \text{if } L_{ij}^E(t) > 0, \\ 0, & \text{else.} \end{cases} \quad (\text{Equation 28})$$

Further, the acquired output activities of such nodes are ranked in the order to constitute an excitatory matrix E_2 . Similarly, the OFF channel extracts the inhibitory activities from the above inhibitory activity matrix I_1 but inhibits those excitatory activities. Specifically, after acquiring the output of the counterpart in the matrix, its node (i, j) outputs an inhibitory activity $L_{ij}^{OFF}(t)$ given by

$$L_{ij}^{OFF}(i, j, t) = \begin{cases} L_{ij}^I(t), & \text{if } L_{ij}^I(t) < 0, \\ 0, & \text{else.} \end{cases} \quad (\text{Equation 29})$$

All such activities are arranged in the order to form an inhibitory matrix I_2 .

M-layer

The medulla layer consists of two parallel sub-layers with $M \times N$ nodes per layer distributed in a matrix, i.e., excitatory (E) and inhibitory layers (I). In the excitatory layer, each node (i, j) directly receives the activity of the node (i, j) in the ON channel, i.e., $L_{ij}^{ON}(t)$ in the matrix E_2 . Afterward, it outputs a stimulatory activity by

$$K_L^{ON}(i, j, t) = L_{ij}^{ON}(t) - \sum_{-1 \leq k, l \leq 1} w_{kl} L_{i+k, j+l}^{ON}(t-1), \quad l \neq 1, \quad (\text{Equation 30})$$

where w_{kl} as a left excitatory radius is defined as the element (k, l) of the left excitatory convolution kernel C_{21} given by

$$C_{21} = \begin{bmatrix} 1 & 0.5 & 1 \\ 1.5 & 0 & 1.5 \\ 1 & 0.5 & 1 \end{bmatrix} \quad C_{22} = -C_{21}$$

Similarly, in the inhibitory layer, the node (i, j) collects the activity of the node (i, j) in the OFF channel, i.e., $L_{ij}^{OFF}(t)$ in the matrix I_2 . Subsequently, it generates an inhibitory activity by

$$K_L^{OFF}(i, j, t) = L_j^{OFF}(t) - \sum_{-1 \leq k, l \leq 1} w_{rs} L_{i+r, j+s}^{OFF}(t-1), l \neq 1 \quad (\text{Equation 31})$$

where w_{rs} denotes the left inhibitory radius defined as the element (k, l) of the left inhibitory convolution kernel C_{22} above.

LDN

LDN is exploited to orderly gather the output activities from the excitatory and inhibitory sublayers in the M-layer, while yielding two global direction activities to characterize the changes of the left visual movements in the field of view, i.e., $\rho_L^{ON}(t)$ and $\rho_L^{OFF}(t)$. $\rho_L^{ON}(t)$ is decided by the behavioral output of the excitatory layer in the ON channel, namely

$$\rho_L^{ON}(t) = 2\text{Sig}\left(\sum_{i=1}^M \sum_{j=1}^N K_L^{ON}(i, j, t)\right) - 1, \quad (\text{Equation 32})$$

with the sigmoid function $\text{Sig}(\cdot)$. Conversely, $\rho_L^{OFF}(t)$ denotes the output of the inhibition layer in the OFF channel, given by

$$\rho_L^{OFF}(t) = 2\text{Sig}\left(\sum_{i=1}^M \sum_{j=1}^N K_L^{OFF}(i, j, t)\right) - 1. \quad (\text{Equation 33})$$

It is highlighted that, as a feedforward visual neural network, L-DVNN outputs a pair of activities at any moment t to characterize the change of visual movement in the left direction. When the LSGO problem in the section of ‘[problem background and motivation](#)’ is solved, L-DVNN executes visual information processing by virtue of the difference of the grayscale images which correspond to the t -th and $(t-1)$ -th state/candidate solution matrices respectively, and later, outputs a pair of activities $(\rho_L^{ON}(t), \rho_L^{OFF}(t))$ to characterize the change of the t -th visual movement. On the other hand, such activities named left learning rates guide states in the t -th state matrix to move along the left direction in terms of the state update strategy, to seek the optimal solution omnidirectionally.

Other neural subnetworks

The R-, U- and D-DVNNs share the designs of the P-, R- and L-layers in the L-DVNN. The R-DVNN utilizes (Equation 30) and (Equation 31) to generate stimulatory and inhibitory activities in its M-layer, but requires that the formula of $l \neq 1$ be replaced by $l \neq -1$. Similarly, so do the U- and D-DVNNs by replacing $l \neq 1$ with $k \neq -1$ and $k \neq 1$, respectively. Besides, the R-, U- and D-DVNNs output their excitatory and inhibitory activities through their neurons RDN, UDN and DDN, respectively. RDN generates a pair of output activities (i.e. $\rho_R^{ON}(t)$ and $\rho_R^{OFF}(t)$) after replacing L in (Equation 32) and (Equation 33) with R . Likewise, U- and D-DVNNs produce their pairs of output activities $(\rho_U^{ON}(t), \rho_U^{OFF}(t))$ and $(\rho_D^{ON}(t), \rho_D^{OFF}(t))$ after replace L as U and D in the two formulas, respectively.

Summarily, DV-NN performs the process of visual information processing by synchronously operating the L-, R-, U-, and D-DVNNs. To characterize the change of visual movement in the field of view, it outputs four pairs of global motion-directional activities so-called global learning rates by means of the four neural subnetworks, i.e., $(\rho_L^{ON}(t), \rho_L^{OFF}(t))$, $(\rho_R^{ON}(t), \rho_R^{OFF}(t))$, $(\rho_U^{ON}(t), \rho_U^{OFF}(t))$, and $(\rho_D^{ON}(t), \rho_D^{OFF}(t))$. Within an iterative cycle, the process, which DVNN generates the four pairs of global learning rates, can be summarized by Algorithm 1 below.

In the L-DVNN, the P-layer as the input layer does not execute any arithmetic or logical operation. The R-layer performs MN subtraction operations. The L-layer runs $36MN$ arithmetic and logical operations, since it needs to generate two activity matrices through the ON and OFF channels. The M-layer receives excitatory and inhibitory activities from the two channels in the L-layer, and then performs lateral inhibition, which needs to $24MN$ operations of addition and subtraction. Further, LDN as an output neuron is required to produce two global left motion direction activities by Equations 4.8 and 4.9, which needs to operate $2(MN+4)$ arithmetic operations. Therefore, L-DVNN needs $63MN + 8$ arithmetic and logical operations to implement the process of visual information processing. Besides, the R-, U-, and D-DVNNs involve in the P-, R-, and L-layers as in the L-DVNN, being all required to compute respective stimulatory and inhibitory activity matrices in their M-layers as well as the outputs of their neurons. Hence, each of them operates $26MN + 8$ arithmetic operations. Thereby, Algorithm 1 runs $141MN + 32$ times within an iterative period, and hence the DVNN’s complexity is $O(141MN)$.

Example 4.1 Take the minimization example with the object function of $\sum_{i=1}^2 [x_i^2 - 10 \cos(2\pi x_i) + 10]$ and the decision domain of $[-20, 20] \times [-20, 20]$ for example to formulate the flowchart of generating the L-DVNN’s global left ON and OFF motion direction activities.

The grayscale image in Figure 8A gives a graphical representation of the objective function with the minimum around 10.0. Take L-DVNN for example to exhibit the process of visual information processing. Figures 8B–8E indicates that, after performing the P-, R-, and L-layers, L-DVNN generates two excitatory and inhibitory matrices to measure the intensive change of the current visual movement, whereas Figures 8F and 8G characterizes the changes of the local motion directions of the current states. Besides, the two global left motion directions, presented in Figures 8F–8G) by the thick arrows guides the current states’ transition.

Based on the dragonfly visual information mechanism as in the previous section, this section explicitly discusses the DVNN’s topological structure and computational model per neural sublayer. The network outputs four pairs of on-line learning rates by Algorithm 1 that will guide the DVNN’s state transition in the next section.

Algorithm 1. The DVNN's algorithm formulation

Input: $\mathbf{X}^{(t)}, f(\mathbf{X}^{(t)})$, $M \times N$. %The state matrix and the related grayscale image
Output: $\rho_k^{ON}(t), \rho_k^{OFF}(t)$, $k \in \{L, R, U, D\}$. % Output four pairs of activities or learning rates
1: for every (i, j) do % R-layer
2: Compute $R_{ij}(t)$ by (Equation 25); % Generate an excitatory or inhibitory activity at each node
3: end for.
4: for every (i, j) do % L-layer
5: Decide $L_{ij}^E(t)$ by (Equation 26); % LMC-sublayer/excitation feature extraction
6: Determine $L_{ij}^I(t)$ by (Equation 27); % LMC-sublayer/inhibition feature extraction
7: Calculate $L_{ij}^{ON}(t)$ and $L_{ij}^{OFF}(t)$ by (Equation 28) and (Equation 29); % RTC-sublayer/ON and OFF-channels
9: end for.
10: for every (i, j) do % M-layer
11: Decide $K_L^{ON}(i, j, t)$, $K_L^{OFF}(i, j, t)$ by (Equation 30) -(Equation 31); %L-DVNN
12: Compute $K_R^{ON}(i, j, t)$, $K_R^{OFF}(i, j, t)$; %R-DVNN
13: Determine $K_U^{ON}(i, j, t)$, $K_U^{OFF}(i, j, t)$; %U-DVNN
14: Calculate $K_D^{ON}(i, j, t)$, $K_D^{OFF}(i, j, t)$; %D-DVNN
15: Compute $K_D^{ON}(i, j, t)$, $K_D^{OFF}(i, j, t)$; %D-DVNN
16: end for.
17: Decide $\rho_L^{ON}(t)$, $\rho_L^{OFF}(t)$ by (Equation 32) - (Equation 33); % Learning rates in the left direction
18: Determine $\rho_R^{ON}(t)$, $\rho_R^{OFF}(t)$; % Learning rates in the right direction
19: Compute $\rho_U^{ON}(t)$, $\rho_U^{OFF}(t)$; % Learning rates in the upper left direction
20: Calculate $\rho_D^{ON}(t)$, $\rho_D^{OFF}(t)$.

Dragonfly visual evolutionary neural network

This section will explicitly emerge the DVENN's topological structure and algorithmic formulation as well as its computational complexity, in which a new state update strategy is developed to execute state transition.

Recall that DVENN, schematically illustrated by Figure 9 is composed of two sequentially connected DVNN and the state update layer, in which DVNN plays a role in guiding state transition. The state update layer includes two sublayers. One is the candidate state update layer in which each node propagates four candidate states by means of four pairs of global learning rates produced by the four output neurons in DVNN, and the other is the candidate selection layer in which the state at each node is decided by the four candidate states at the matched node in the last layer.

State update

Candidate state update

The layer comprises $M \times N$ nodes displayed in a matrix form. At each node, the four candidate states i.e., left state $\mathbf{x}_L^{(t)}$, right state $\mathbf{x}_R^{(t)}$, upper state $\mathbf{x}_U^{(t)}$, and down state $\mathbf{x}_D^{(t)}$ are generated randomly when $t = 1$. Conversely, each of them is updated in terms of four pairs of global learning rates. More precisely, related to the inspiration of differential evolution, the left state $\mathbf{x}_L^{(t)}$ is updated into $\mathbf{x}_L^{(t+1)}$ by means of $\rho_L^{ON}(t)$ and $\rho_L^{OFF}(t)$, namely

$$\mathbf{x}_L^{(t+1)} = \mathbf{x}_L^{(t)} + \zeta \rho_L^{ON}(t) (\mathbf{x}_{gb} - \mathbf{x}_{r1}^{(t)}) + \eta \rho_L^{OFF}(t) (\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}), \quad (\text{Equation 34})$$

with random variables $\zeta, \eta \in U(0, 1)$, where $\mathbf{x}_{r1}^{(t)}$, $\mathbf{x}_{r2}^{(t)}$, and $\mathbf{x}_{r3}^{(t)}$ are randomly picked up from the state set of $\{\mathbf{x}_L^{(t)}, \mathbf{x}_R^{(t)}, \mathbf{x}_U^{(t)}, \mathbf{x}_D^{(t)}\}$, and $\mathbf{x}_L^{(t)}$ is the current state at the matched node in the candidate selection layer at time t . Similarly, $\mathbf{x}_R^{(t)}$, $\mathbf{x}_U^{(t)}$ and $\mathbf{x}_D^{(t)}$ are orderly transformed into $\mathbf{x}_R^{(t+1)}$, $\mathbf{x}_U^{(t+1)}$ and $\mathbf{x}_D^{(t+1)}$ by

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \zeta \rho_k^{ON}(t) (\mathbf{x}_{gb} - \mathbf{x}_{r1}^{(t)}) + \eta \rho_k^{OFF}(t) (\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}), \quad (\text{Equation 35})$$

with $k \in \{R, U, D\}$.

Eqs.(Equations 34 and 35) indicate that the role of the candidate state update layer is to derive four new candidate states at each node in terms of four pairs of global learning rates and the differential evolution-like state update strategy. Herein, to discover a more potential solution from different directions, the four pairs of global learning rates as state transition amplitudes guide the related state to transform into four new candidate states.

Candidate selection

The layer, with the same structure as that in the candidate state update layer, is to update all the current states in $\mathbf{X}^{(t)}$ in terms of the acquired candidate states in the candidate state update layer. Precisely, the state $\mathbf{x}^{(t)}$ at each node is updated by the best one among $\mathbf{x}_L^{(t)}$, $\mathbf{x}_R^{(t)}$, $\mathbf{x}_U^{(t)}$, and $\mathbf{x}_D^{(t)}$. This can be achieved by

Algorithm 2. The state update layer's algorithm formulation

Input: $X^{(t)}$, $\rho_k^{ON}(t)$, $\rho_k^{OFF}(t)$, $k \in \{L, R, U, D\}$, x_{gb} .

Output: $X^{(t+1)}$.

- 1: for $\forall x^{(t)} \in X^{(t)}$ do
- 2: Compute $x_L^{(t+1)}$, $x_R^{(t+1)}$, $x_U^{(t+1)}$, and $x_D^{(t+1)}$; % (Equation 34) - (Equation 35)
- 3: Calculate $f(x_k^{(t+1)})$ with $k \in \{L, R, U, D\}$;
- 4: Decide $x^{(t+1)}$; % (Equation 36)
- 5: end for.
- 6: Decide the best state x_{pb} in $X^{(t+1)}$.
- 7: if $f(x_{pb}) < f(x_{gb})$ then
- 8: $x_{gb} \leftarrow x_{pb}$;
- 9: end if.

$$x^{(t+1)} = \operatorname{argmin}\{f(x_L^{(t+1)}), f(x_R^{(t+1)}), f(x_U^{(t+1)}), f(x_D^{(t+1)})\}. \quad (\text{Equation 36})$$

Summarily, related to the designs of the above two sub-layers, the state update layer can be implemented by Algorithm 2.

In the above state update layer, the candidate state update sub-layer needs to create four candidate states at each node with $20D+8$ arithmetic operations, and thus performs $MN(20D+8)$ operations. Each new candidate state is evaluated with m_p operations. In addition, the candidate selection sub-layer needs to operate $3MN$ operations in deciding the states of the MN nodes, while determining the best state x_{pb} with $MN-1$ comparisons between $X^{(t+1)}$ and x_{gb} . Thus, within an iterative period, Algorithm 2 execute C_q arithmetic operations with $C_q = 4MN(5D + m_p + 11) - 1$, and hence the complexity is $O(4MN(5D + m_p))$.

Algorithm formulation on DVENN

Related to the formulations of Algorithms 1 and 2, the D-VENN's algorithm is given by Algorithm 3.

It is emphasized that Algorithm 3 presents the whole process of solving the LSGO problem as in the section of 'problem background and motivation', depending Algorithms 1 and 2. Therein, at the initial step, Algorithm 3 takes the difference of two grayscale images $f(X^{(0)})$ and $f(X^{(1)})$ as the DVNN's input. After that, within an iterative period, on the one hand, it is required to acquire four pairs of learning rates in terms of the DVNN, on the other hand DVENN is requested to generate a new state matrix $X^{(t)}$ at the moment t and update x_{gb} , relying upon such learning rates and Algorithm 2. Related to Algorithms 1, 2, and 3 executes M_r arithmetic operations within an iterative loop. Herein,

$$M_r = 141MN + 32 + C_q.$$

The formula shows that the complexity of Algorithm 3 is $O(4MN(5D + m_p))$, due to $C_q = 4MN(5D + m_p + 11) - 1$. Therefore, it is a fast optimizer in that M and N are required to take small values.

Algorithm 3. DVENN's algorithm

Input: $M \times N$, G_{max} , C_{kl} , $k, l = 1, 2$.

Output: x_{gb} .

- 1: $t = 1$.
- 2: Initialization:
Randomly initialize state matrices: $X^{(0)} = (x_{ij}^{(0)})_{M \times N}$ and $X^{(1)} = (x_{ij}^{(1)})_{M \times N}$; % $x_{ij}^{(0)}, x_{ij}^{(1)} \in D$
Compute $f(x_{ij}^{(0)})$ and $f(x_{ij}^{(1)})$ named grayscale values, $1 \leq i \leq M, 1 \leq j \leq N$;
Take the best state in $X^{(1)}$ as x_{gb} ; % $x_{gb} = \operatorname{argmin} f(X^{(1)})$
- 3: while $t \leq G_{max}$ do
Execute Algorithm 1 to produce $\rho_k^{ON}(t)$, $\rho_k^{OFF}(t)$, $k \in \{L, R, U, D\}$; %Output four pairs of learning rates
Implement Algorithm 2: %Generate new states by the candidate state update strategy.
Decide $X^{(t+1)}$ and x_{gb} ; %Update $X^{(t)}$ and x_{gb} by the candidate selection strategy
 $t = t + 1$.
- 4: end while

QUANTIFICATION AND STATISTICAL ANALYSIS

In this experimental study, all experiments were conducted on a computer with Intel Core(TM)i5-9400F CPU/2.90GHz and 16.0 GB RAM, using the Visual C++ platform. For all methods, the solution search process was terminated when the fitness evaluation exceeded -3×10^6 . Each method was run 25 times on each example in the CEC 2010 and CEC2013 benchmark test suites. Through the validation results of the CEC2010 and CEC2013 benchmark test sets, DVENN achieved absolute victory in comparison with 7 other comparative swarm intelligence optimization algorithms. DVENN can converge to a global or approximate global solution at a relatively fast speed.

For engineering application problems, this paper takes the standard two-stage operational amplifier as an example to simplify the parameter design of analog integrated circuits. With the goal of maximizing open-loop low-frequency gain, we abstracted mathematical optimization models. Also, we applied the DVENN algorithm and other seven comparison algorithms to practical engineering optimization problems, and conducted model simulation experiments in the Empyrean Aether (Huada Jiu Tian) simulation environment to achieve the optimization of various performance indicators of the circuit. The experimental results have demonstrated the practical application significance of the DVENN algorithm in engineering application problems.