

Multi-Swarm Particle Swarm Optimization Co-Evolution Algorithm based on Principal Component Analysis for Solving Conditional Nonlinear Optimal Perturbation

Li Zhao^{1, a}

¹School of Software Engineering, Tongji University, Shanghai, 201804, China

^aemail: sanpizhaoli@126.com

Keywords: PSO; Co-Evolution; CNOP; ZC Model

Abstract. Conditional nonlinear optimal perturbation (CNOP) is an initial perturbation evolving into the largest nonlinear evolution at the prediction time. It has played an important role in predictability and sensitivity studies of nonlinear numerical models. Generally, the solution for CNOP is the spectral projecting gradient algorithm which is based on the adjoint model. However, many numerical models have no corresponding adjoint models and new implementations of these adjoint models cost tremendous engineering work. In this paper, we propose a multi-swarm PSO co-evolution algorithm based on principal component analysis to solve CNOP. To demonstrate the validity, the Zebiak-Cane model is utilized as a case to verify the proposed method. Experimental results show that the proposed method can be treated as an approximate solution to CNOP.

Introduction

Conditional nonlinear optimal perturbation (CNOP) [1] is an initial perturbation evolving into the largest nonlinear evolution at the prediction time. It has played an important role in predictability and sensitivity studies of nonlinear numerical models. CNOP has been applied to analyze the optimal precursor of El Nino-Southern Oscillation (ENSO) [2]. The common method for solving CNOP is the spectral projecting gradient (SPG2) [3]. It requires two inputs: the objective function and corresponding gradient which is obtained by adjoint models. Unfortunately, many modern numerical models have no corresponding adjoint models, and it is a huge work to implement a new one. Thus, it is necessary to develop new approaches without gradient.

Evolutionary algorithms (EAs) [4] have been used to solve CNOP. Wen [5] proposed simulated annealing based ensemble projecting method to solve CNOP. The genetic algorithm was applied by Zhang [6] for solving CNOP in Zebiak-Cane model. However, there is only one population in their methods. As individual population algorithms could easily fall into the local optimum, multiple populations can be a better choice.

Co-evolution was proposed by Potter [7], which simulates the evolution process of an ecosystem consisting of two or more species. In the case of simulating optimization processes, a number of populations are generated and evolve concurrently. During evolution time, they can share information with others.

As the solution space of modern numerical model usually has extremely high dimensions, dimensional reduction methods could be used. Principal component analysis (PCA) [8] is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. In this paper, we propose utilizing multi-swarm PSO co-evolution algorithm [9] based on principal component analysis to solve CNOP of practical models, and perform the case study of the Zebiak-Cane model which is famous for the successful forecast of the ENSO event.

CNOP

CNOP is the perturbation $u_{0\delta}^*$ that make the objective function $J(u_0)$ obtain the maximum in the condition of $\|u_0\| \leq \delta$, i.e.

$$J(u_{0\delta}) = \max \|J_{TE}\|^2, \quad s.t. \|u_0\|_A \leq \delta \quad (1)$$

$$J_{TE} = M_{t_0 \rightarrow t}(\overline{u_0} + u_0) - M_{t_0 \rightarrow t}(u_0) \quad (2)$$

Where $M_{t_0 \rightarrow t}$ is the propagator of a nonlinear model from the initial time t_0 to the prediction time t . J_{TE} is the nonlinear evolution of the initial perturbation, u_0 is the initial perturbation and the δ is the magnitude of uncertainty. Eq. 1 could also be transformed into a minimum problem as follows:

$$J(u_{0\delta}) = \min(-\|J_{TE}\|^2), \quad s.t. \|u_0\| \leq \delta \quad (3)$$

Thus, the gradient of J could be obtained as follows:

$$\nabla J = -2 * H^T * J_{TE} \quad (4)$$

Where H is the tangent matrix of the nonlinear model $M_{t_0 \rightarrow t}$, and H^T is called the Jacobean matrix of J_{TE} . To compute H^T , corresponding adjoint models have been developed and used. However, it is difficult for modern numerical models that without adjoint models to solve CNOP.

Multi-Swarm PSO Co-Evolution Algorithm based on PCA

In previous study [9], there is conclusion that a driven dissipative system can get into a steady state which is constitutive of attractors with low dimension. And CNOP can be projected onto the space expanded by the attractors. Therefore, evolutionary algorithm could be utilized to obtain CNOP in low-dimensional space. The PCA based multi-PSO co-evolution algorithm (MPSOCE) mainly has two steps: the feature extraction and multi-swarm PSO co-evolution.

Feature Extraction.

To get the training data set X , we have ran the model from initial state. The extracted features have statistical meaning which is similar with the character of attractors. The principal components can be acquired by the eigen-decomposition:

$$XX^T P = P \Sigma \quad (5)$$

Where Σ is the diagonal matrix whose diagonal entries correspond to the eigenvalues of XX^T . As XX^T is positive definite matrix, the columns of the obtained eigenvectors P are mutually orthogonal. Then take the k ($k \ll n$) eigenvectors corresponding to the top k biggest eigenvalues.

Multi-swarm PSO Co-Evolution.

Particle swarm optimization (PSO) [10], is one of the most effective optimization algorithms. It is described as:

$$v_i^{t+1} = \omega * v_i^t + c_1 * r_1 * (pbest_i^t - x_i^t) + c_2 * r_2 * (gbest^t - x_i^t) \quad (6)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (7)$$

Where ω is an inertia weight coefficient, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random numbers in the range[0,1]. x_i is the location of the i -th particle on the t -th iteration, v_i denotes a velocity vector of the i -th particle on the t -th iteration. $pbest_i$ is personal best location that gives the best value of the evaluation function of the i -th particle on the i -th iteration, $gbest_i$ is global best location that gives the best value of the evaluation function on the i -th iteration. Based on the idea of co-evolution, we propose a new movement formula of each particle:

$$v_i^{t+1} = \omega * v_i^t + c_1 * r_1 * (pbest_i^t - x_i^t) + c_2 * r_2 * (Swarm_gbest^t - x_i^t) \quad (8)$$

$Swarm_gbest^t$ means the location that gives the best value of the evaluation function on the i -th iteration of all swarms. To population S_i ($1 \leq i \leq n$) in each iteration, while the particles are evolving, some poor particles will be selected to be killed in certain ratios. Specifically, a pair of parent particles is selected for breeding child particle from the pool selected previously. And child particles will replace the dead ones. To x_{ij}^t , that is the j dimension of particle x_i^t at t -th iteration, we have:

if ($random(0.0,1.0) < property$)

$$x_{ij}^t = parent_{1j}$$

else

(9)

$$x_{ij}^t = \text{parent}_{2j}$$

Where parent_1 and parent_2 are the parent particles, and fit_1 and fit_2 are their fitness function values. The new child particle will replace dead particle if it meets the threshold, otherwise the process repeats until satisfaction, we set the *property* as follows:

$$\text{property} = \text{fit}_1 / (\text{fit}_1 + \text{fit}_2) \quad (10)$$

At the end of iteration, some elite particles of each swarm will be selected to form gene pool in ratio. A gene particle of the pool and a non-genetic particle will be chosen to breed new child particle, which will substitute the non-genetic particle. The process is similar with (9). We have algorithm 1:

Algorithm 1: MPSOCE algorithm	
Input:	training data set X
1:	Make eigen-decomposition take top k eigenvectors
2:	while the termination is not satisfied do
3:	Randomly generate initial populations N and particles S
4:	For each particle $i = 1, 2, \dots, S$ do
5:	Compute adaption value $f(x_i)$
6:	Select particles with larger value to form gene pool
7:	Update each particle's velocity, position, adaption value
8:	End do
9:	Select some poor particles of each swarm to be killed
10:	Form gene pool of all swarms
11:	Pick gene particles and non-genetic particles to breed new ones
12:	End while
Output:	the optimal solution CNOP

Experiments

To show the validity of the MPSOCE algorithm, we apply it to solve CNOP in the Zebiak-Cane (ZC) model [11]. The ZC model successfully forecasted the El Nino event of 1986-1987. Since then, it has been widely applied to the research of the predictability and dynamics of ENSO. Moreover, we perform a comparison between MPSOCE and the adjoint-based method, which is called XD method [2]. Due to XD method's high precision, it is usually treated as the benchmark. In this paper, we compare the magnitudes and patterns of CNOP generated by the two methods. Some recently proposed methods like SAEP and PCAGA have also be compared with our methods.

ZC model.

The ZC model is a mesoscale air-sea coupled model for the Tropical Pacific, and composed of three sub-models: the atmospheric model, ocean model and coupled model. In ZC model, u_0 in Eq. 2 is a perturbation vector consists of the sea surface temperature anomalies (SSTA) and thermocline depth anomalies (THA).

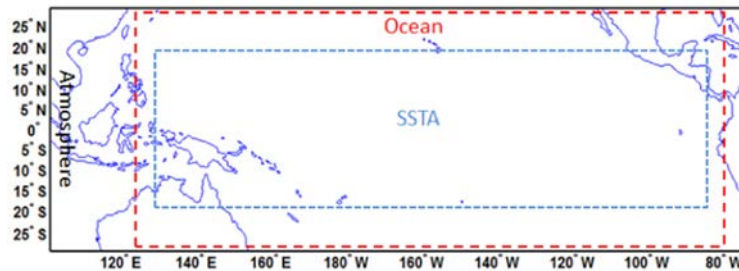


Fig. 1 Region of ZC model

The region of ZC coupled model is shown in figure 1. The inner rectangle is the integration region of SSTA in the model. The middle rectangle is the region of the ocean model. The outer rectangle represents the region of the atmospheric model. The training data set is acquired through the integration of the ZC model over 200 years. The dimension of ZC model grid is 30*34 (for

SSTA and THA respectively). Due to the fact that there is no value in the marginal area, the marginal area is removed to reduce the size of samples in the training set to $20 \times 27 \times 2$.

Comparison Analysis.

According to previous study [6], we set eigenvectors k as 80. In [12], we take their recommended parameters of PSO. We take ω as 0.729, $c_1=c_2=1.49$.

We set each month as the initial month and its optimization time span as 9 months, then run two methods. The magnitude of CNOP in each initial month is shown in figure 1. It is obvious that all lines have the same trend. The magnitudes of MP SOCE are little less than SAEP in April, but greater than it in other months. Moreover, it has greater magnitudes than PCAGA and MP SOCE. Difference between the XD method and MP SOCE is shown in table 1. The biggest difference is in February while the smallest one in April. Without loss of generality, we choose the two months as the examples to show the pattern comparison.

In figure 2, the left column is the results from MP SOCE and the right is from XD method. Three rows of sub-figures are the patterns of the SSTA, THA and the SSTA evolution. In the first row, it is obvious that the left part is negative while the right part is positive, which is the character of El Nino. In the third row, the patterns of SSTA evolution in both methods look similar. Although the THA in the second row look different, they both have the characteristic that middle areas are positive while other areas are negative. Since MP SOCE just extracts part of all eigenvectors, its THA pattern looks smoother than that in XD method. And the THA pattern could better reflects the range of ocean current in east pacific. The result of April is shown in figure 3. The CNOP pattern generated by MP SOCE looks similar with XD method in both cases that have the biggest and smallest magnitude difference. Therefore, we could draw the conclusion that MP SOCE can be treated as an approximate solution for CNOP.

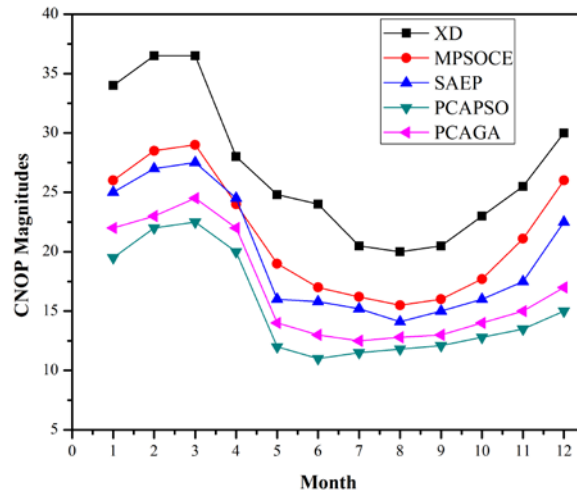


Fig. 2 The magnitude of CNOP of 5 methods

In Fig 3, the left column represents the results from MP SOCE and the right is from the XD method. Three rows of sub-figures are the patterns of the SSTA, THA and the SSTA evolution. In the first row, the left SSTA keep the main large-scale characters of the right one. It is obvious that the left part is negative while the right part is positive, which is the character of El Nino. In the third row, the patterns of SSTA evolution in both methods look similar except that the left one is a little weaker than the right one. Although the THA in the second row look different at first glance, they both have the characteristic that middle areas are positive while other areas are negative. Since MP SOCE just extracts part of all eigenvectors, its THA pattern looks smoother than that in XD method. And the THA pattern could better reflects the range of ocean current in east pacific.

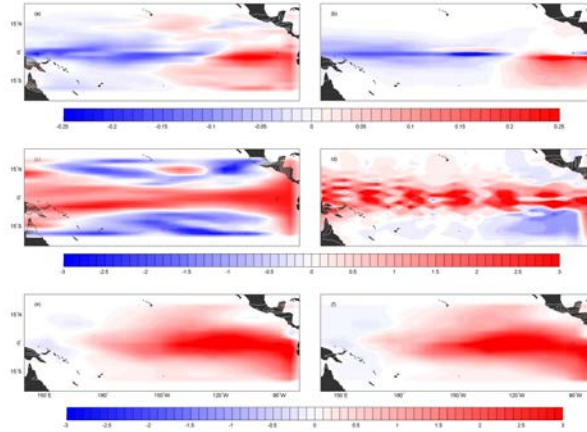


Fig. 3 CNOP pattern in February

As the magnitudes of CNOP in February from both methods have biggest difference, the previous results in April can also be observed, which is shown in figure 4. In other words, the CNOP pattern generated by MPSOCE looks similar with that by the XD method in both cases which have the biggest and smallest magnitude difference. Therefore, we could draw the conclusion that MPSOCE can be treated as an approximate solution for CNOP.

Table 1. The difference between XD method and MPSOCE

Method \ Month	1	2	3	4	5	6	7	8	9	10	11	12
XD	34.0	36.5	36.5	28.0	24.8	24.0	20.5	20.0	20.5	23.1	25.5	30.0
MPSOCE	26.0	28.4	29.0	24.0	19.0	17.0	16.2	15.5	16.0	17.7	21.1	25.8
diff	8.0	8.1	7.5	4.0	5.8	7.0	4.3	5.5	4.5	5.4	4.4	4.2

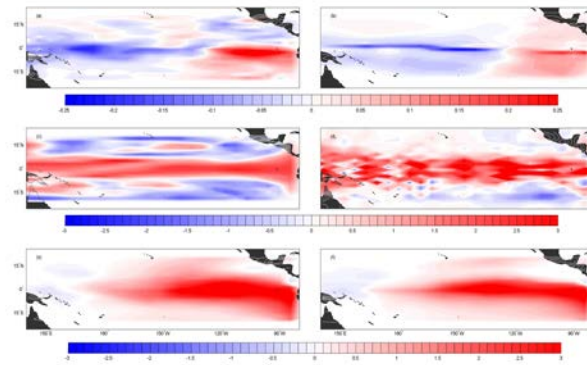


Fig. 4 CNOP pattern in April

Efficiency Analysis.

In Table 2, when 80 eigenvectors are taken, the proposed method consumes about 14.9 minutes while the XD method needs 13.4 minutes with 30 initial guess fields(IGF), and the time PCAGA and MPSOCE need is 21 minutes and 18.3 minutes respectively.

Table 2. Time consumption of five methods

Method	Time (mins)
XD (1 IGF)	0.45
XD (30 IGF)	13.4
MPSOCE	13.8
SAEP	14.9
PCAGA	21.0
MPSOCE	18.3

Generally, the XD method is much faster than our method when it takes only one initial guess

field. But the XD method maybe obtains the locally optimum, which means the global optimum cannot be obtained in this case. Thus we use 30 initial guess fields in this case. Therefore, the proposed method has similar computational efficiency with the XD method and has better efficiency than other algorithms.

Conclusion

This paper proposes a multi-swarm PSO co-evolution algorithm based on principal component analysis to solve CNOP. PCA is applied to the data set to obtain principal components. The proposed method is applied to search CNOP on the space. Compared with other proposed algorithms, our method has better CNOP magnitude and time efficiency. Although our method just has similar running time with the XD method, our method is free of adjoint model. That is, we can apply the proposed method to solve CNOP in other numerical models without adjoint models, which could improve the capability of the applications of CNOP.

Acknowledgment

In this paper, the research was sponsored by the Youth Foundation of National Natural Science Fund of China (Project No.61401173 and No.41405097).

References

- [1] Mu Mu, Wansuo Duan, A new approach to studying ENSO predictability: Conditional nonlinear optimal perturbation[J]. Chinese Science Bulletin. 2003, 48, 1045-1047.
- [2] Wansuo Duan, Mu Mu, Bin Wang, Conditional nonlinear optimal perturbation as the optimal precursors for El Nino Southern oscillation events[J]. Geography Research. 2004, 109, 1-12.
- [3] EG. Birgin, JEM. Martinez, M. Raydan, Nonmonotone Spectral Projected Gradient Methods on Convex Sets[J], Society for Industrial and Applied Mathematics Journal on Optimization, 2000, 10, 1196–1211.
- [4] Thomas Back, Handbook of Evolutionary Computation, Bristol, U.K./New York, NY, USA: Inst. Phys. Publishing/Oxford Univ. Press, 1997.
- [5] Shichen. Wen, Shijin Yuan, SAEP: Simulated Annealing Based Ensemble Projecting Method for Solving Conditional Nonlinear Optimal Perturbation[C], Algorithms and Architectures for Parallel Processing, 2014, pp. 655-668.
- [6] Bin. Mu, Linlin. Zhang. PCAGA: Principal Component Analysis Based Genetic Algorithm for Solving Conditional Nonlinear Optimal Perturbation[C]: submitted to International Joint Conference on Neural Networks (2015).
- [7] MA. Potter, KA. De Jong, A cooperative coevolutionary approach to function optimization[C], International Conference of Parallel Problem Solving from Nature, 1994, 2, 249–257.
- [8] AR Osborne, A. Pastorello, Simultaneous occurrence of low-dimensional chaos and colored random noise in nonlinear physical systems[J], Physic Letter. 1993, 181, 159–171.
- [9] Ke Xu, Dong Liu, Algorithm of multi-PSO co-evolution based on GA and PSO[C], Conference of Evolution. 2009, 45, 51-54.
- [10] J. Kennedy, Eberhart, Particle Swarm Optimization, IEEE International Conference of Neural Networks, 1995, 1942-1948.
- [11] Zebiak, A model El Nino-Southern Oscillation, Mon. Wea. Rev. 1987, 115, 2262–2278.
- [12] Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space[C], IEEE Transaction on Evolution Computation. 2002, 6, 58-73.