International Conference on Artificial Intelligence (IC-AI), Las Vegas, USA, 2002: 1163-1169

Solving Numerical Optimization Problems by Simulating Particle-Wave Duality and Social Information Sharing

Xiao-Feng Xie
Institute of Microelectronics,
Tsinghua University
Beijing 100084, P. R. China

Wen-Jun Zhang
Institute of Microelectronics,
Tsinghua University,
Beijing 100084, P. R. China

Zhi-Lian Yang
Institute of Microelectronics,
Tsinghua University
Beijing 100084, P. R. China

Abstract - *The demand for the solutions to different complex numerical optimization problems has long outstripped the ability of engineers to supply them. Since the numerical optimization is a static problem that is naturally similar to the movement of particulates with particle-wave duality in potential field, it can be simulated by the assistance of cooperative searching agents with social information sharing. A paradigm evolution algorithm PACA is realized with few parameters. The experiments by comparing PACA with genetic algorithms (GA) and particle swarm optimization (PSO) on some famous benchmark functions show that it can get high-quality solutions efficiently.*

Keywords - *Evolutionary Algorithm, particle swarm optimization, genetic algorithm*

1. Introduction

The general numerical optimization problem is defined as finding $X \in S \subseteq R^n$ such that [1]

$$\begin{cases} f(X) = \min\{f(Y); Y \in S\}, \\ g_j(X) \le 0,\ for\ j = 1,\dots,m \end{cases} \tag{1}$$

Where f and g_j are functions on S; S is a search space defined as a Cartesian product of domains of variables x_i's ($1 \le i \le$ n). The set of points that satisfying all the constraints g_j is denoted F.

The problem (1) is often treated as finding the minimum value of a single fitness function *eval*(X) with the constraint-handling methods [1]. Since *eval*(X) is static, it is easily to be mapped as a potential energy function $E(X)$ to describe the potential field with the natural comparability. An object that is located at X has the energy $E(X)$. A 1-D example of *eval*(X) is showed in Fig. 1, which X_a, X_b are the local minimum, and X_o is the global optimum points.

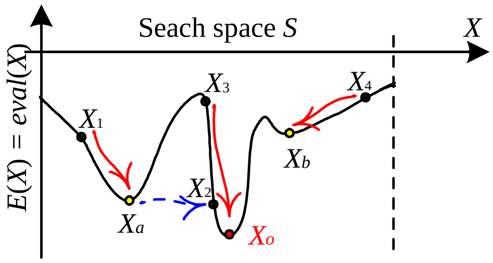

Fig. 1 Fitness function from the viewpoint of potential energy

An classical object is tending to the location with lower potential energy, until reaching the bottom of an valley if no barrier is encountered, such as $X_1 \to X_a$, $X_3 \to X_o$, $X_4 \to X_b$ in Fig. 1. The physical process is directly simulating by hill-climbing method, however, it cannot perform searching efficiently if many valleys exists.

However, for microcosmic particulate, from the particle-wave duality theory [2], when it enters the potential field, on the one hand, it will be induced to the local minimum of a valley due to its particle property; on the other hand, it has the probability to transfer to other lower valleys by traversing the potential barrier due to its wave property, such as $X_a \rightarrow X_2$ in Fig. 1, which is so-called *tunnel effect*. At last, the particulate will be found at the global minimum valley.

The particle-wave duality provides a physical foundation for global numerical optimization. However, it has a major difficulty in terms of contriving a computation simulation for the physical mechanism, with an obvious reason: unclear potential field landscape before sophisticated searching. Fortunately, the agent-based technique [3-6] represents a challenging new approach, which has the potential to operating for the uncertain environment.

In this paper, a new evolutionary algorithm for solving numerical optimization problem is performed by simulating the particle-wave duality, which is realized by the assistance of cooperative searching agents to detect the neighborhood of particulates. A paradigm algorithm PACA (PAticulates & Cooperative Agents) is realized with few parameters. It has also been demonstrated on some benchmark functions to perform efficiently than genetic algorithms with homomorphous mappings in [1] and particle swarm optimization [5].

2. Simulating particle-wave duality

Suppose a potential barrier is quadrate as in Fig. 2, which the height is V_0, the width is a. For a particulate with energy E and effective-mass m, G. Gamow [2] (p. 112) had deduced the tunnel penetration probability P for the case that $V_0 > E$:

$$P(a)=e^{-ka} \tag{2}$$

Where $k = \frac{2}{\hbar}\sqrt{2m(V_0 - E)}$, $\hbar$ is plank constant.

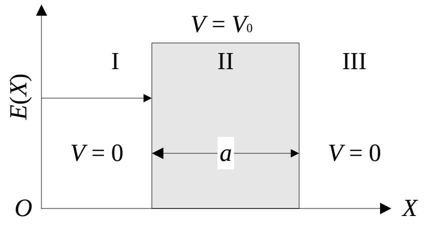

Fig. 2 Tunnel penetration for potential barrier

Since the unclear shape of potential field for computer simulation, an searching agent is employed to detecting the neighborhood of the particulate stochastically with distribution probability $P_a(l)$ for different distance l to the particulate by supposing possible barrier width l. In order to describe the tunnel effect, $P_a(l)$ is set to be in proportion to $P(l)$. For ith dimension, the normalized distance l_i to particulate is varied from 0 to 1. Then the normalized distribution probability of the searching agent $P_a(l_i)$ is

$$P_a(l_i)=cP(l_i) \tag{3}$$

Where $c = \left(\int_0^1 P(l_i) dl_i \right)^{-1}$.

The searching agent may control the particle or wave property of particulate with different k, which is decided by energy difference (V_0-E) and effective-mass m of particulate. For a particulate, a small k means more wave property, i.e. more freedom in potential field, which has more exploration ability while mapping to search process; and a large k means more particle property, i.e. easily to be trapped in a valley, which has more exploitation ability.

3. Social sharing of information

Although a particulate will arrive to the global optimum point at last since it is ergodic in the search space S from the physical foundation, it is obviously that its searching efficiency should still be greatly affected by the starting location X. For example, as in Fig.1, a particulate that initially located in x_2 is more easily reaching to x_o than in $x_{1.}$

Population-based method is a good choice to provide more robustly searching. A population of searching agents is introduced to assist the movement of particulates. Then the problem arises: how to schedule the execute times among searching agents reasonably?

Some rules can be observed underlie animal social behavior, including schools [4], flocks [5], and ants [6], and that of humans. As E. O. Wilson suggests [4], in reference to fish schooling, that social sharing of information among agents offers an evolutionary advantage, which has been make great success [5][6].

4. A paradigm of PACA algorithm

Definition 1: An e*nergy point* is located in search space *S*, which is described by the location *X* and its energy value *E*(*X*). *Information table* is a set of e*nergy points*.

Definition 2: *Searching agent* is an intelligent agent, which is location on an e*nergy point* in *information table*.

Definition 3: For point X_1, *neighborhood searching* for X_2 is selecting a new point *X'*, which for the *i*th dimension of *X'*,

$$X'[i]=X_1[i]+2*rand()*(X_2[i]-X_1[i]) \qquad (4)$$

Where *rand*() is random number in (0,1). Here the X_2 is defined as *central point*, X_1 is defined as *reference point*.

4.1 Simulating combined particulates

For different potential field *E*(*X*), the searching efficiency will be affected by the particle or wave property of particulate. In this sense, maybe an adaptive *k* is necessarily. A compromised choice is combined searching method, which by combining a set of particulates with different *k* that are scheduled by searching agent in different weight *w*, i. e.,

$$P_a'(l_i)=\sum c_j' P_j(l_i) \qquad (5)$$

Where for *j*th particulate, $c_j'=w_j c_j \Big/ \sum w_j$, $P_j(x_i)=e^{-k_j x_i}$.

The equation (5) can be simulated by successive neighborhood searching. A tunnel control coefficient $C_{tc} \in (0,1)$ is predefined. For each searching step, if *rand*()<C_{tc}, the searching agent will move to a new energy point at random; else the location of the particulate is set as a central point, and the location of the searching agent is set as a reference point, and then the searching agent will move to a new energy point in the neighborhood of the particulate base on the neighborhood searching according to the equation (4).

Fig. 3 gives the distribution probability $P_a'(l)$ of the searching agent with the normalized distance x_j to the particulate (by testing with totally 1E8 times successive neighborhood searching in 1-D space for C_{tc} = 0.2), which can be described well by a three-order exponential decay function, i.e. three combined particulates with different *k* and *w*.

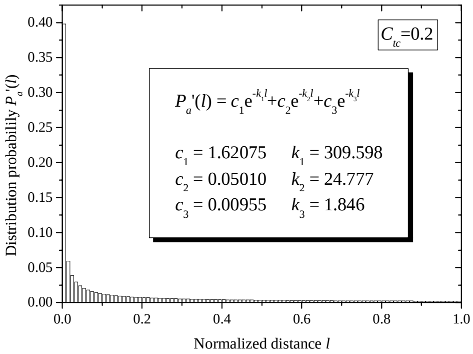

Fig. 3 The distribution probability of searching agent with the normalized distance to the particulate (for C_{tc}=0.2)

4.2 Simulating cooperative agents

With an information table shared by all agents, the local information sharing is incorporated by using the tournament selection model [7]. Two or more energy points are picked at random from the information table and are compared based on their potential energy value, which the point with lower energy is chosen as the current location of a particulate.

4.3 Algorithm realization

The principle for PACA paradigm is shown in Fig. 4. Where the particle-wave duality in nature system is simulated by combined particulates with the agent to ensure the ergodicity and the cooperated agents in social system is simulated by information sharing among agents to enhance the robusticity. The connection of the systems is completed by successive neighborhood searching.

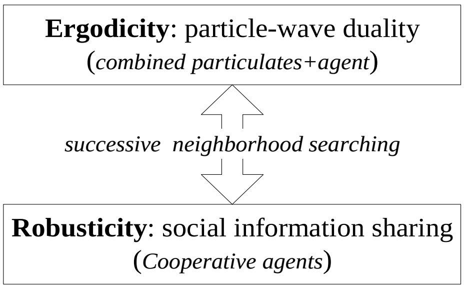
Ergodicity: particle-wave duality
(*combined particulates+agent*)
successive neighborhood searching
Robusticity: social information sharing
(*Cooperative agents*)

Fig. 4 Principle diagram for PACA

Suppose the number of energy points in information table is N, the number of searching agents is m, the realization of PACA is:

a) Initialize all the N energy points (include the location X and its $E(X)$) in information table;

b) Allocate each searching agent to occupy an energy point in information table at random;

c) For each searching agent, if $rand()<C_{tc}$, then the searching agent will move to a new energy point in the search space S at random; else the located energy point of the searching agent is set as reference point, then another energy point in information table is selected as the location for a particulate based on tournament selection from two (or more) energy points, which is set as central point, and the agent will move the a new energy point base on the neighborhood searching;

d) Replacing the m energy points with highest energy values (i.e. the worst points) in information table by the m new energy points;

e) Repeat the previous c)–e) step until a stop condition (for example, a predefined generation number T) is reached.

The parameters of PACA include: number of energy points in information table N, number of searching agents m, tunnel control coefficient C_{tc}, and maximum number of generations T. The total number of function calculations is $T_e = N+m*T$.

5. Experimental setting

The test functions in [1] for constrained numerical optimization problems are proposed by Z. Michalewicz et al. These test cases include objective functions of various types (linear, quadratic, cubic, polynomial, etc.) with various numbers of variables and different types (linear inequalities, nonlinear inequalities, etc.) and numbers of constraints. The topologies of feasible search space are also quite different. These test cases are summarized in table 1. For each test case we list number n of variables, type of the function f, and the relative size of the feasible region in the search space given by the ratio ρ, which was determined experimentally by generating 1E6 random points from S and checking whether they belong to F, the number of constraints of each category (linear inequalities LI, nonlinear equations NE and inequalities NI), and the number a of active constraints at the optimum (including equality constraints).

Table 1: Summary of test cases [1]

Func.	n	Type of f	ρ	LI	NE	NI	a
G_1	13	quadratic	0.0111%	9	0	0	6
G_2	50	nonlinear	99.8474%	0	0	2	1
G_3	50	polynomial	0.0000%	0	1	0	1
G_4	5	quadratic	52.1230%	0	0	6	2
G_5	4	cubic	0.0000%	2	3	0	3
G_6	2	cubic	0.0066%	0	0	2	2
G_7	10	quadratic	0.0003%	3	0	5	6
G_8	2	nonlinear	0.8560%	0	0	2	0
G_9	7	polynomial	0.5121%	0	0	4	2
G_{10}	8	linear	0.0010%	3	0	3	6
G_{11}	2	quadratic	0.0000%	0	1	0	1

This paper has not discussion the G_3, G_5, G_{11} in [1], which has almost 0% feasible space due to the equations constraints, since it needs to replace the equations constraint $g(X)$=0 by an inequality constraint $|g(X)| \leq \varepsilon$ for some small ε >0, which the testing results will not be comparable since ε is not clearly defined in [1].

5.1 Algorithm setting for GAs in [1]

The GAs in [1] is based on Gray coding, and incorporated proportional selection, function scaling, and standard operators. All parameters were fixed: *Pop_size* = 70, *generation gap* = 100%, and p_c = 0.9. The only non-standard feature was a variable probability of mutation: $p_m(t) = p_m(0) - (p_m(0) - p_m(T)) * (t/T)^r$, where t and T are the current and maximum generation numbers, respectively. In all experiments, $p_m(0)$ = 0.005, r = 4, and $p_m(T)$ = 0.00005.

The experiment #1 of GAs in [1] was executed 20 runs. For each run the maximum number of generations was set to T = 5000 (For G_2, T = 10000), and for each run a random reference point was selected (i.e. the first randomly generated feasible point was accepted as a reference point). The total evaluation times is equal to *Pop_size***gap***T*=350000 (For G_2, is 700000) during evolution stage.

The employed constraint-handling method is homomorphous mapping [1], which establishes a one-to-one mapping between feasible space F and the n-dimensional cube. Since the infeasible space is taken out from S, which will reduce the difficulty for optimization that other constraint-handling methods. However, in the same time, the homomorphous mapping method will resulted in an average two-fold increase of computational time [1] due to a binary search involved to establish intersection points of a line segment and the boundaries of the feasible part of the search space.

5.2 Algorithm setting for PSO [5]

PSO is an evolutionary algorithm, which is inspired by swarm intelligence. Each potential solution, call *particle*, is also assigned a randomized velocity. The location of the *i*th particle is represented as $X_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$, where $x_{id} \in [l_d, u_d]$, $d \in [1, D]$, l_d, u_d are the lower and upper bounds for the *d*th dimension, respectively. The best previous position (which giving the best fitness value) of the *i*th particle is recorded and represented as $P_i = (p_{i1}, \dots, p_{id}, \dots, p_{iD})$, which is also called *pbest*. The index of the best *pbest* among swarm is represented by the symbol g. The location P_g is also called *gbest*. The velocity for the *i*th particle is represented as $V_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$, is clamped to a maximum velocity $V_{max} = (v_{max,1}, \dots, v_{max,d}, \dots, v_{max,D})$. The PSO concept consists of, at each time step, changing the velocity and location of each particle toward its *pbest* and *gbest* locations according to the equations (6a) and (6b):

$$v_{id}=w*v_{id}+c_1*rand()*(p_{id}-x_{id})+ c_2*rand()*(p_{gd}-x_{id}) \quad (6a)$$

$$x_{id}=x_{id}+v_{id} \quad (6b)$$

Where w is inertia weight, c_1 and c_2 are acceleration constants, and *rand*() is a random function in the range [0, 1]. The parameters of PSO includes: number of particles m = 14, a linearly decreasing w which from 0.9 to 0.4, c_1=c_2=2, v_{max}=0.5(u_d-l_d). PSO was executed 50 runs. For each run T = 1000 (For G_2, T = 20000). The total evaluation times is equal to m*T=14000 (For G_2, is 280000).

5.3 Algorithm setting for PACA

PACA is based on real-valued coding. All parameters were fixed: N = 70, m = 14, C_{tc} = 0.01. PACA was executed 100 runs. For each run T = 1000 (For G_2, T = 20000). The total evaluation times is equal to N+m*T=14070 (For G_2, is 280070).

Table 2: Comparison for the test results between PACA, GAs in [1], and PSO

Func	Optimum attribution		PACA			Experiment #1 of GAs in [1]			PSO	
	type	value	Worst	Best	Average	Worst	Best	Average	Average	P_f (%)
G_1	Min.	-15	-12.9995	-15.0000	-14.9379	-14.0566	-14.7207	-14.4609	-5.0200	0
G_2	Max.	0.803553	0.74419	0.80335	0.78873	0.78427	0.79506	0.79176	0.54663	0
G_4	Min.	-30665.538	-30663.444	-30665.537	-30665.454	-30617.0	-30662.5	-30643.8	-30592.68	0
G_6	Min.	-6961.814	-6953.381	-6961.789	-6960.593	-4236.7	-6901.5	-6191.2	-6960.97	94
G_7	Min.	24.306	26.928	24.424	24.965	38.682	25.132	26.619	1201.981	12
G_8	Max.	0.095825	0.095825	0.095825	0.095825	0.029143	0.095825	0.087155	0.095825	2
G_9	Min.	680.63	681.2474	680.6401	680.7493	682.88	681.43	682.18	680.9443	4
G_{10}	Min.	7049.33	8624.071	7085.576	7381.212	11894.5	7215.8	9141.7	12177.14	24

The constraint-handling method for PSO and PACA is following the criteria [8]: a) any feasible solution is preferred to any infeasible solution; b) among two feasible solutions, the one having better objective function value is preferred; c) among two infeasible solutions, the one having smaller constraint violation is preferred.

6. Results and discussion

Table 2 gives the comparison for test results of PACA, GAs in [1], and PSO. Where for PSO, P_f gives the percentage out of all trails that is failed in entering feasible space, and only successful trails are counted for average results.

It can be found that PACA provides better results for almost all the cases. Moreover, PACA shows robustness than PSO in almost same evaluation times, especially for G_6. And for all cases, the total evaluation times of PACA are much less than that of GAs in [1].

There has an important problem we met with the GAs: premature convergence. After a few runs of the algorithm, the population tends to be too homogeneous, and the action of crossover is not sensible. The fitness of the individual that far from the convergent space is often worse than the fitness of most individuals in current population. Even the individual is created by the action of mutation, it is often deprived the breed right according to natural selection. In this stage, the GA performs as a purely random searching algorithm by the mutated individuals. The searching becomes highly inefficiently since even the probability of mutation is often small.

By considering the equation (5) as similar to the probabilistic distribution for mutation operation, the PACA algorithm is similar to evolutionary strategies (ES) with hybrid social information sharing in searching agents. In ES, each individual is mutated by adding a random function (the most commonly used random function either a Gaussian or Cauchy function) [9], while in PACA each searching agent is updated according to the simulating particle-wave duality. The wave property of particulate will ensure the exploration ability during the whole searching process. Even the energy points in information table are similar to each others, the agent will move to new energy point that far from the convergent space with the tunnel effect according to the wave property. Then the cooperative agents will explore from the unclear space to convergent space according to the nonlinear interactions by social information sharing with considerable efficiency.

7. Conclusions

PACA is a simple algorithm that simulating particle-wave duality property of particulates in potential field base on the natural comparability for this kind of static field for numerical optimization problems. The particle property of particulate is emphasizing particularly on locating the minimum of local valley quickly, while the wave property of particulate is providing the probability that traversing to the lower valley through tunnel effect.

A paradigm of PACA is realized. Much further research remains to be conducted on this simple paradigm. The goals in developing it have been to keep it simple and robust, and it seems to have succeeded at that. The algorithm has few parameters, and the experiments for the numerical optimization problems typically used in evolutionary optimization research show that PACA provides better solutions in less evaluation times, while comparing with the GAs in [1] and PSO.

References

[1] S. Koziel, Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *IEEE Transaction on Evolutionary Computation*, Vol. 7, pp. 19-44, 1999

[2] Fu-Jia Yang, *Atomic physics*, Higher education press, Beijing, China, 2000

[3] P.Cristea, A. Arsene, R. Nitulescu, "Evolutionary intelligent agents," *Proc. of the Congress on Evolutionary Computation*, pp. 1320-1328, 2000

[4] E. O. Wilson, *Sociobiology: the new synthesis*, Belknap Press, MA, 1975

[5] R. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," *Proc. IEEE Int. Conf. on Evolutionary Computation*, pp. 81-86, 2001

[6] E. Bonabeau, M. Dorigo, G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, Vol. 406, pp. 39-42, 2000

[7] D. B. Fogel, *Evolutionary computation: towards a new philosophy of machine intelligence*, IEEE Press, NJ, 1996

[8] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Engrg.*, Vol. 186, pp. 311-338, 2000

[9] X. Yao, Y. Liu, "Fast Evolution Strategies," *Control and Cybernetics*, Vol. 26(3), pp. 467-496, 1997