Asynchronous Parallel Evolutionary Algorithms for Constrained Optimizations

Kang Li-shan, Liu Pu, Kang Zhuo, Li Yan, Chen Yu-ping
State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
National Laboratory for Parallel and Distributed Processing

Abstract: Recently Guo Tao proposed a stochastic search algorithm in his PhD thesis for solving function optimization problems. He combined the subspace search method (a general multi-parent recombination strategy) with the population hill-climbing method. The former keeps a global search for overall situation, and the latter keeps the convergence of the algorithm. Guo’s algorithm has many advantages, such as the simplicity of its structure, the higher accuracy of its results, the wide range of its applications, and the robustness of its use.

In this paper a preliminary theoretical analysis of the algorithm is given and some numerical experiments has been done by using Guo’s algorithm for demonstrating the theoretical results. Three asynchronous parallel evolutionary algorithms with different granularities for MIMD machines are designed by parallelizing Guo’s Algorithm.

Key words: asynchronous parallel evolutionary algorithm; function optimization

CLC number: TP 301.6

0 Introduction

There are efficient and effective algorithms for solving function optimization, such as genetic algorithms and evolution strategies\[^{[1,2]}\]. Recently Guo Tao\[^{[3]}\] proposed a new population random search algorithm which is based on the subspace search (a general multi-parent recombination search strategy) combining with population hill-climbing for solving function optimization problems with inequality constraints. Numerical experiments show that Guo’s Algorithm is very efficient and effective. But there is a little theoretical analysis of the algorithm, that is unfavourable for its application. In this paper, the authors try to give a preliminary theoretical analysis of the algorithm and based on which three asynchronous parallel algorithms are designed with different granularities for MIMD machines, such as shared memory multiprocessors and message passing multicomputers.

1 GUO’s Algorithm and Its Theoretical Analysis

For the sake of simplicity, we consider the following function optimization problem:

\[ \min_{X \in D} f(X) \]  \hspace{1cm} (1)

where \( f(X) \) is the objective function, \( X = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n \) and \( D = \{ X | l_i \leq x_i \leq u_i, i = 1, 2, \ldots, n \} \).

Select \( m \) points \( X_i, i = 1, 2, \ldots, m \) from \( P \) to form a subspace:

\[ V = \{ X \in D | X = \sum_{i=1}^{m} a_i X_i \} \]

where \( \sum_{i=1}^{m} a_i = 1 \), and \( -0.5 \leq a_i \leq 1.5 \).

The Guo’s algorithm is described as follows.

ALGORITHM GUO-1:

Begin

Received date: 2000-04-29
Foundation item: Supported by the National Natural Science Foundation of China (No. 70071042, 60073043), the National 863 Hi-Tech Project of China (No. 883-306-2T06-06-3) and the National Laboratory for Parallel and Distributed Processing.
Biography: Kang Li-shan (1934-), male, Professor, research interests: parallel computing and evolutionary computation. E-mail: kang @whu.edu.cn
initialize \( P = \{X_1, X_2, \ldots, X_N\} \), \( X_i \in D \)
\[ t \leftarrow 0 \]
\[ X_{\text{best}} = \arg \min_{1 \leq i \leq N} f(X_i) \]
\[ X_{\text{worst}} = \arg \max_{1 \leq i \leq N} f(X_i) \]
While \( f(X_{\text{best}}) \neq f(X_{\text{worst}}) \) do
\[ \text{select } m \text{ points } X_1, X_2, \ldots, X_m \text{ from } P \text{ randomly;} \]
\[ \text{select one point } X' \text{ from } V \text{ randomly;} \]
If \( f(X') < f(X_{\text{worst}}) \) then
\[ t \leftarrow t + 1 \]
\[ X_{\text{best}} = \arg \min_{1 \leq i \leq N} f(X_i) \]
\[ X_{\text{worst}} = \arg \max_{1 \leq i \leq N} f(X_i) \]
end do
output \( t, P \);
end

where \( N \) is the population size and \( m - 1 \) is the dimension of subspace \( V \) if the vectors \( X_i, i = 1, 2, \ldots, m \) are linear independent. \( t \) is the number of iterations. \( X_{\text{best}} = \arg \min_{1 \leq i \leq N} f(X_i) \) denotes one of the arguments (individuals) which satisfy \( f(X_{\text{best}}) = \min_{1 \leq i \leq N} f(X_i) \).

Remark 1: INITIALIZE \( P \) means selecting \( N \) points (individuals) from \( D \) randomly to form an initial population.

Remark 2: \( N \) can be selected according to the dimension \( n \) of the problem and the complexity of its landscape. If \( n \) is large and \( f(X) \) is complex, then \( N \) is large too. In our experiments \( N \) is chosen as \( 20 \leq N \leq 150 \).

Remak 3: To our experiments, \( m \) is chosen as \( 7, 8, 9 \) or \( 10 \).

Remark 4: For solving function optimization problems with inequality constraints:
\[ \min_{X \in D^*} f(X) \tag{2} \]
where \( D^* = \{X \in D; g_i(x) \leq 0, i = 1, 2, \ldots, q\} \).

Denote \( h_i(x) = \begin{cases} 0, & g_i(x) \leq 0 \\ g_i(x), & \text{otherwise} \end{cases} \)
and \( H(x) = \sum_{i=1}^{q} h_i(x) \).

Define a logic function:
\[ \text{better}(X_1, X_2) = \begin{cases} H(X_1) \leq H(X_2) & \text{true} \\ H(X_1) > H(X_2) & \text{false} \\ (H(X_1) = H(X_2)) \land (f(X_1) \leq f(X_2)) & \text{true} \\ (H(X_1) = H(X_2)) \land (f(X_1) > f(X_2)) & \text{false} \end{cases} \]
to express that \( X_1 \) is better than \( X_2 \).

Then Guo's algorithm for solving problem (2) can be described as follows.

ALGORITHM GUO-2:

Begin
initialize \( P = \{X_1, X_2, \ldots, X_N\} \), \( X_i \in D \)
\[ t \leftarrow 0 \]
\[ X_{\text{best}} = \arg \text{better}(X_{\text{best}}, X), \forall X \in P \]
\[ X_{\text{worst}} = \arg \text{better}(X, X_{\text{worst}}), \forall X \in P \]
while not better(\( X_{\text{worst}}, X_{\text{best}} \)) do
\[ \text{select } m \text{ points } X_1, X_2, \ldots, X_m \text{ from } P \text{ randomly;} \]
\[ \text{select one point } X' \text{ from } V \text{ randomly;} \]
If better(\( X', X_{\text{worst}} \)) then\( X_{\text{worst}} = X' \)
\[ t \leftarrow t + 1 \]
\[ X_{\text{best}} = \arg \text{better}(X_{\text{best}}, X), \forall X \in P \]
\[ X_{\text{worst}} = \arg \text{better}(X, X_{\text{worst}}), \forall X \in P \]
end do
output \( t, P \);
end

where \( X_{\text{best}} = \arg \text{better}(X_{\text{best}}, X), \forall X \in P \) denotes the variable \( X_{\text{best}} \in P \) which satisfies better(\( X_{\text{best}}, X \), \( \forall X \in P \).

A preliminary theoretical analysis of Guo's algorithm is as follows:

1) The algorithm adopts a population search strategy as the evolutionary algorithms, that is useful for finding the global optimum.

2) The algorithm adopts a stochastic search strategy in the random subspaces, especially the searches in the subspace are not convex recombination:
\[ x' = \frac{m}{i=1} a_i x_i', \sum_{i=1}^{m} a_i = 1 \]
and
\[ -0.5 \leq a_i \leq 1.5 \]
such that the search subspace can cover the multi-parent recombination space and guarantee the ergodicity of stochastic searches. It means that in the solution space there is no room which can not be searched by the algorithm.

3) The algorithm adopts the selection strategy: elimination of the worst. In every iteration, only the worst individual in the population is eliminated. In this case, the selection pressure is minimum. It ensures the diversity of the population and the long life of the fine individuals in the population. This is a population hill climbing strategy...