Neurocomputing 132 (2014) 68–78

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A human–computer cooperative particle swarm optimization based immune algorithm for layout design

Fengqiang Zhao [a,b], Guangqiang Li [a,*], Chao Yang [a], Ajith Abraham [c,d], Hongbo Liu [a,c,**]

[a] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
[b] College of Electromechanical and Information Engineering, Dalian Nationalities University, Dalian 116600, China
[c] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, WA 98071, USA
[d] IT4 Innovations - Center of excellence, VSB - Technical University of Ostrava, Ostrava 70833, Czech Republic

ARTICLE INFO

Article history:
Received 18 October 2012
Received in revised form
20 February 2013
Accepted 8 March 2013
Available online 1 December 2013

Keywords:
human–computer cooperation
Genetic algorithms
Immune function
Particle swarm optimization
Hybrid
Layout

ABSTRACT

Packing and layout problems have wide applications in engineering practice. However, they belong to NP (non-deterministic polynomial)-complete problems. In this paper, we introduce human intelligence into the computational intelligent algorithms, namely particle swarm optimization (PSO) and immune algorithms (IA). A novel human–computer cooperative PSO-based immune algorithm (HCPSO-IA) is proposed, in which the initial population consists of the initial artificial individuals supplied by human and the initial algorithm individuals are generated by a chaotic strategy. Some new artificial individuals are introduced to replace the inferior individuals of the population. HCPSO-IA benefits by giving free rein to the talents of designers and computers, and contributes to solving complex layout design problems. The experimental results illustrate that the proposed algorithm is feasible and effective.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Packing and layout problems [1,2] deal with how to put objects into a limited space reasonably under given constraints. These constraints include the requirements for equilibrium, stability, connectivity and adjacent states. Some methods [1,3] are presented, such as mathematical programming and criterion methods, heuristic algorithms, graph theory, expert systems and natural laws. But it is still difficult to solve the problems satisfactorily.

Unlike those traditional methods, the swarm intelligent algorithms illustrated more superior performances in recent years [4–7]. They are particularly fit for solving medium or large-scale problems [8] but for the complex packing and layout problems, there exist some defects, such as premature convergence and slow convergence rate. In this paper, we first introduce the novel strategies into our hybrid PSO-based immune algorithm (PSO-IA). In addition, an intelligent machine or an algorithm is powerful in numerical calculation and repetitive operations but lack of experience and inspiration, which are just the strong points of human beings. There are very differences in nature between computer and human, which also mean that they should deeply depend on each other when solving practical complex problems. The idea of man–machine synergy (or called as human–computer cooperation) was originated by Lenat and Feigenbaum [9]. It is regarded as a promising approach to solve complex engineering problems [10,11]. According to this idea, taking into account the intractable nature of the packing and layout problems and their importance, we further propose a novel human–computer cooperative algorithm based on PSO-IA (HCPSO-IA).

2. Hybrid PSO-based immune algorithm

In this section, we present a hybrid PSO-IA algorithm with some improvement strategies on parallel GA (PGA) [12], which mainly include immunity principle, new PSO update operators, arithmetic-progression rank-based selection with pressure as well as a multi-subpopulation evolution based on improved adaptive crossover and mutation. PSO-IA is also the basis of our human–computer cooperative algorithm.

2.1. Arithmetic-progression rank-based selection with pressure

In genetic algorithms, a rank-based selection model focuses on the numerical size relations rather than the specific numerical

* Corresponding author.
** Corresponding author at: School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China.
E-mail addresses: fqzhao2002@163.com (F. Zhao), gqlimail@163.com (G. Li), yangchaoneu@sina.com (C. Yang), ajith.abraham@ieee.org (A. Abraham), hongboliu.ucsd@gmail.com, lhb@dlut.edu.cn (H. Liu).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.03.062

differences among individual fitness values. A probability assignment table has to be preset. But there is no deterministic rule for design of the table. Moreover, it is difficult to make the selection probabilities of individuals adaptively changed along with evolution process [13,14]. We introduce arithmetic-progression rank-based selection with pressure based on the mathematical concept of interpolation method.

There is one independent parameter in this operator, selection pressure α. It denotes the ratio of the maximal individual selection probability P_{max} to the minimal one P_{min} within a generation, i.e. $P_{max}=\alpha P_{min}$. It numerically shows the superiority that the better individuals are reproduced into the next generation during selection operation and it is changeable along with algorithm evolution. In the early stage, lesser α can maintain population diversity and prevent the algorithm from premature convergence, while in the late stage, greater α can benefit accelerating convergence. Let $\alpha=f(K)$, K denotes the generation number. Assume that α_{max} and α_{min} denote the maximum and minimum of selection pressure respectively, then

$$\alpha=\frac{(K-1)(\alpha_{max}-\alpha_{min})}{K_{max}-1}+\alpha_{min} \tag{1}$$

where K_{max} is the maximal generation number set in algorithm. And our numerical experiments and statistical analysis show that α_{max} and α_{min} may be chosen in the interval [6, 15] and [1.5, 5] respectively [15].

To calculate the selection probability of every individual, we arrange all the individuals within a population in descending order based on their fitness values. Let Ind_i represent the ith individual within a population as well as F_i and P_i represent its fitness and selection probability respectively. There exist Ind_i $(i=1,2,\ldots,M)$ and $F_i>F_{i+1}$ $(i=1,2,\ldots,M-1)$. M is the population size. Suppose that the selection probability values of all the individuals form an arithmetic progression. Its first term and last term are $P_1=P_{max}=\alpha P_{min}$ and $P_M=P_{min}$ respectively. Obviously, the sum of all the individual selection probability is 1, i.e. subtotal of arithmetic progression as follows:

$$\begin{aligned} S_M &= [(P_1+P_M)\cdot M]/2 \\ &= [(\alpha P_M+P_M)\cdot M]/2 \\ &= 1 \end{aligned} \tag{2}$$

We get

$$P_M=2/[M(1+\alpha)] \tag{3}$$

Therefore we obtain the common difference of the arithmetic progression

$$\begin{aligned} \Delta &= (P_1-P_M)/(M-1) \\ &= [P_M(\alpha-1)]/(M-1) \\ &= [2(\alpha-1)]/[M(1+\alpha)(M-1)] \end{aligned} \tag{4}$$

And there exists

$$\begin{aligned} P_i &= P_1-(i-1)\Delta \\ &= \alpha P_M-(i-1)\Delta \end{aligned} \tag{5}$$

Substituting Eqs. (3) and (4) into Eq. (5), it is easy to find that

$$P_i=\frac{2\alpha\cdot(M-i)+2(i-1)}{M\cdot(\alpha+1)(M-1)}, \quad i=1,2,\ldots,M \tag{6}$$

In the process of selection, we firstly reproduce the best individual of current generation and have its copy in the next generation directly based on the elitist model, and then figure out selection probabilities of all individuals according to Eq. (6) and finally generate the remaining $M-1$ individuals of the next generation by the fitness proportional model. Compared with traditional rank-based selection, the advantage of proposed selection operator is that it can conveniently change the selection probabilities of individuals by changing selection pressure.

2.2. Antibody concentration and immune selection

Immunity-based algorithms that originated in 1990s have many good characteristics [16]. They can embody immune memory, extraction and inoculating efficient antibodies as well as antibody inhibition and promotion mechanism in the biological immune systems. So the immunity strategy is helpful to prevent evolutionary algorithms from premature convergence and accelerate convergence rate [17,18]. In this section, we introduce immune principle into parallel genetic algorithms and put forward some improvements as follows:

- We adopt the simple and easy Euclidean distance to calculate affinities between antibodies (i.e. individuals) for convenient to engineering design.
- We present correction formula for calculating individual concentration and the immune selection operator based on above proposed arithmetic-progression rank-based selection with pressure.
- We propose the individual migration strategy according to the immune memory mechanism between subpopulations in the hybrid algorithm. The parallel GA provides a paradigm for our multi-class (multi-subpopulation) evolution. (For details, see Section 2.4 .)

2.2.1. Antibody affinity and antibody concentration

Here antibodies are exactly individuals. They have the same concept and all represent solutions of a given problem. Antibody affinity ay_{vw} defined as follows indicates similar extent between antibody v and antibody w.

$$ay_{vw}=1/[1+H(2)] \tag{7}$$

The range of ay_{vw} is within $(0,1]$. If the value ay_{vw} is higher then the antibody v is more similar with antibody w. At present, $H(2)$ in last formula is mostly calculated by average information entropy formula based on antibody v and w. In fact, as above stated, antibody affinity denotes similar extent between antibodies. In other words, $H(2)$ represents the distance between two antibodies. It can be calculated by average information entropy and also can be calculated by other methods, if two conditions are satisfied. One is $H(2)\geq 0$, and $H(2)=0$ indicates that the genes of two antibodies are exactly the same. The other is that greater differences between genes of two antibodies can lead to greater value of $H(2)$. In order to simplify the calculation and be easy for real-number coding and engineering realization, we adopt Euclidean distance to calculate affinities.

Let antibody $\vec{v}=(v_1,v_2,\ldots,v_n)$ and antibody $\vec{w}=(w_1,w_2,\ldots,w_n)$, then

$$H(2)=\sqrt{\sum_{i=1}^{n}(v_i-w_i)^2} \tag{8}$$

If M denotes the population size, concentration c_v of antibody v in its population is usually defined by Eq. (9).

$$c_v=\frac{1}{M}\sum_{w=1}^{M}ay_{vw} \tag{9}$$

Obviously there exists $c_v\in(0,1]$.

In order to avoid oscillation during the later period of proposed algorithm and facilitates algorithm convergence, antibody concentration c_v has to tend to 1 ultimately along with an increase in the value of generation number K. Therefore, we present a correction for Eq. (9) as follows:

$$C_v=\left(\frac{1}{M}\sum_{w=1}^{M}ay_{vw}\right)^{(1-K/K_{max})\beta} \tag{10}$$

where β is a system parameter and usually set β=0.5.

Download English Version:

https://daneshyari.com/en/article/406552

Download Persian Version:

https://daneshyari.com/article/406552

Daneshyari.com