Optik 125 (2014) 897–902

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

A hybrid membrane evolutionary algorithm for solving constrained optimization problems

Xiao Jianhua [a,*], Huang Yufang [b], Cheng Zhen [c], He Juanjuan [d], Niu Yunyun [e,*]

[a] Logistics Research Center, Nankai University, Tianjin 300071, China
[b] College of Mathematics, Southwest Jiaotong University, Chengdu 610031, China
[c] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[d] Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[e] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 9 April 2013
Accepted 12 August 2013

Keywords:
P systems
Membrane evolutionary algorithm
Particle swarm optimization
Constrained optimization problems

ABSTRACT

Solving constrained optimization problems (COPs) is a central research topic in the field of optimization. Given the complexity of COPs, it is difficult to solve them with traditional optimization techniques. In this paper, a hybrid membrane evolutionary algorithm (HMEA) is proposed. It combines a one-level membrane structure with a particle swarm optimization (PSO) local search algorithm. The simulation results show that the proposed algorithm is valid and outperforms the state-of-the-art algorithms.

1. Introduction

In real-world applications, difficult problems in science and engineering optimization can be characterized as constrained optimization problems. Generally, the constrained optimization can be formulated as follows:

$$Min\, y = (f(X)), \quad X = (x_1, x_1, \ldots, x_1)$$
$$s.t. \begin{cases} g_p(X) \le 0 & p = 1, 2, \ldots, N_g \\ h_q(X) = 0 & q = 1, 2, \ldots, N_h \\ x_{iL} \le x_i \le x_{iU} & i = 1, 2, \ldots, n \end{cases} \tag{1}$$

where $f(X)$ is an objective function defined on E^n, $X \in E^n$ is an n-dimensional real vector that satisfies the restrictions while minimizing the objective function, $g_p(X)$ is the p-th inequality constraint, N_g is the total number of the inequality constraints, $h_q(X)$ is the q-th equality constraint, and N_h is the total number of the equality constraints, and x_{iL} and x_{iU} are the lower and upper bounds of the variable x_i, respectively.

Methods for solving COPs are a very important topic in the field of optimization. However, given the complexity of COPs, it is difficult to solve them using traditional optimization techniques. In recent years, evolutionary algorithms have demonstrated superior performance in both benchmark functions and real-world applications. Various evolutionary algorithms have been developed to solve COPs. In 2006, Ji et al. [1] developed an improved simulated annealing algorithm for solving linear constrained optimization problems. Wang et al. [2,3] proposed a multi-objective optimization method based on hybrid evolutionary algorithms to solve COPs. Karaboga and Akay [4] proposed a modified artificial bee colony (ABC) algorithm for COPs. Inspired by previous work [3], Wang and Cai [5] developed a hybrid multi-swarm particle swarm optimization method that incorporated differential evolution to solve constrained optimization problems. Zou et al. [6] proposed a novel modified differential evolution algorithm for constrained optimization problems. Wang et al. [7] incorporated a novel adaptive computational chemotaxis into the bacterial foraging algorithm (ABTA) to solve COPs. Gao et al. [8] proposed a modified harmony search (HS) method to solve the constrained optimization problems. In 2012, Boussaid et al. [9] proposed a viable stochastic optimization algorithm based on biogeography for constrained optimization problems. Oh et al. [10] presented a new approach for effectively handling COP constraints. This method combined a widely used EA with a stochastic ranking evolutionary strategy.

Membrane computing (i.e., P systems) was initiated by Gheorghe Paun [11] in 1998. This group of algorithms is a class of powerful computing models that are abstracted from the

* Corresponding authors.
E-mail addresses: jhxiao@nankai.edu.cn (J. Xiao), niuyunyun1003@163.com (Y. Niu).

0030-4026/$ – see front matter
http://dx.doi.org/10.1016/j.ijleo.2013.08.032

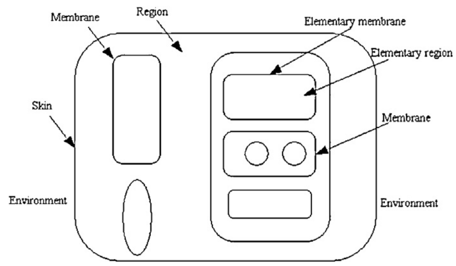

Fig. 1. The membrane structure of a cell-like P system.

structure and functioning of living cells as well as from the interactions between living cells in tissues or higher-order biological structures. Until now, research on membrane computing has mainly concentrated on two topics. The first topic is universality. In this research area, various P systems inspired from living cells have been constructed, and their computational power has been investigated (for example [12–15]). The second topic is efficiency. To theoretically solve various NP-complete problems, a number of P systems were constructed, such as P systems, to solve the PSPACE-complete problem [16], 0–1 knapsack problem [17], tripartite matching problem [18], Hamilton path problem [19] and the maximum clique problem [20].

In recent several years, membrane computing and evolutionary algorithms have been developed; these are nature-inspired models used to solve complex problems. The first attempt in this direction was by Nishida [21,22], who developed membrane algorithms based on the nest membrane structure to solve the traveling salesman problem. After Nishida, Huang et al. [23] proposed a membrane algorithm combining the conventional genetic algorithm to solve single- and multi-objective numerical optimization problems. Furthermore, Huang et al. [24] used an optimization procedure based on the framework of membrane computing to design the optimal controller for marine diesel engines. Leporati et al. [25] developed a polynomial time membrane algorithm that computed approximate solutions to minimum storage problems. A quantum-inspired evolutionary algorithm based on P systems was also developed to solve the knapsack problem [26], the satisfiability problem [27], the image processing [28], and the radar emitter signals problem [29]. In 2012, Xiao et al. [30] developed a membrane evolutionary algorithm to solve the DNA encoding optimization problem. Zhang et al. [31] proposed a hybrid approach based on differential evolution and tissue membrane systems to solve constrained manufacturing parameter optimization problems. The similarities between distributed evolutionary algorithms and P systems have been analyzed, and new variants of distributed evolutionary algorithms have been proposed and applied to some continuous optimization problems [32]. In this paper, a hybrid membrane evolutionary algorithm based on PSO is proposed to solve COPs.

This paper is organized as follows. In Section 2, the basic ideas underlying P systems and PSO are introduced. Section 3 proposes a hybrid membrane evolutionary algorithm (HMEA) based on PSO for constrained optimization problems. The simulation results and analyses are provided in Section 4. Section 5 presents the conclusion and remarks on future research.

2. The basic ideas of P systems and PSO

2.1. P systems

P systems can be classified into three categories: cell-like P systems, tissue-like P systems and neural-like P systems [11]. The membrane structure of a cell-like P system, shown in Fig. 1, consists of several membranes arranged in a hierarchical structure inside a main membrane, called the skin. A membrane that contains no other membranes inside it is said to be elementary. A space delimited by one membrane and the membrane immediately below it is called a region, and the region of an elementary membrane is the space delimited by it. Each region can contain a multiset of objects and a set of evolution rules (according to which objects can evolve) as well as communication rules (according to which objects can move between regions).

A cell-like P system with an output set of objects based on evolution and communication rules is formally defined as follows:

$$\Pi = (O, u, L_1, \ldots, L_n, R_1, \ldots, R_n, i_0)$$

where

(i) O is the alphabet of objects.
(ii) μ is a membrane structure consisting of n membranes with membranes (and hence regions) injectively labeled with $1, 2, \ldots, n$; $n \geq 1$ is denoted as the degree of system $\prod$.
(iii) $L_i (1 \leq i \leq n)$ are sets of strings over O, while L_i is initially placed in region i.
(iv) $i_0 \in \{1,2,\ldots,n\}$ is the label of the output membrane.
(v) $R_i (1 \leq i \leq n)$ are finite sets of evolution rules over O^*. R_i is associated with region i of μ, and it is defined by the following forms:
   (a) $[_i s_1 \to s_2]_i$, where $i \in \{1, 2, \ldots, n\}$, and $s_1, s_2 \in O^*$.
   (Evolution rules: a rule of this type works on a string objects using the local search algorithm or various evolution operators, and the new strings object are created in region i.)
   (b) $s_1 [_i]_i \to [_i s_2]_i$, where $i \in \{1, 2, \ldots, n\}$, and $s_1, s_2 \in O^*$.
   (Send-in communication rules: a string object is sent into region i.)
   (c) $[_i s_1]_i \to [_i]_i s_2$, where $i \in \{1, 2, \ldots, n\}$, and $s_1, s_2 \in O^*$.
   (Send-out communication rules: a string object is sent out of region i.)

P systems, which are considered computational models, are also membrane algorithms. Membrane algorithms are comprised of a sequence of transitions between the configurations of a given system Π, starting from the initial configuration $(L_i, \ldots, L_n)$. In the computing process, the system moves from one configuration to another one by applying the rules associated with regions in a non-deterministic and maximally parallel manner. The rules involving both transformation and communication are responsible for evolving current objects and transferring them among regions according to some targets. The final configuration is produced when the system halts, at which point no more rules are applicable in any region. The result of the computation is a multiset of objects, and it is obtained in region i_0 [27]. A previous study provides more details about P systems [11].

In the standard membrane algorithm, the crossover operation is used as an evolution rule, but this can lead the algorithm to become trapped at a local optional solution at a later period. In this paper, we use PSO to update the string object during the computation process. The basic idea behind PSO is introduced in following section.

2.2. Particle swarm optimization

PSO is an effective optimization method that belongs to the category of swarm intelligence methods, originally developed by Kennedy and Eberhart [33]. During optimization, each particle in the swarm is propelled toward the optimal point by increased velocity. In the original formulation of PSO, the velocity v_{ij}^t and