

# Searching Minimal Fractional Graph Factors by Lattice Based Evolution

XIYU LIU<sup>1</sup>, YINGHONG MA<sup>2</sup>, YANGYANG ZHENG<sup>3</sup>

<sup>1,2,3</sup>School of Management and Economics,

Shandong Normal University, Jinan Shandong, 250014, China

<sup>1</sup>sdxyliu@163.com;sdxyliu@beelink.com;xyliu@sdu.edu.cn

<sup>2</sup>yinghongma71@163.com

<sup>3</sup>zhengyy0727@163.com

*Abstract:* Spanning subgraphs is necessary for the communication in networks. Apart from theoretical existence results, effective technique of searching graph factors is an important problem in graph theory, complex networks, and applications which is NP hard. In this paper, we first propose a lattice based evolution technique. Then we present an evolutionary searching paradigm for the minimum fractal graph factors. A simple Markov analysis of the proposed genetic algorithm is given together with some experiments on the effects of parameters to the performance of the algorithms.

*Key-Words:* Niche, evolution, lattice, minimum factors, graph

## 1 Introduction

Evolutionary computing is the collective name for a range of problem-solving techniques based on the principles of biological evolution, among which genetic algorithm is one of the most important and well-known implementations, see [5][11][13][16] for example. For years, evolutionary computing has found wide applications to a variety of problems ranging from optimization and search problems. Various generalizations of evolutionary computing have emerged including such new techniques as swarm intelligence. However, among evolution's many basic features, an essential one is that the evaluation is based on a scalar valued fitness function. This feature restricts more extensive applications of evolution because on certain circumstances, it is difficult to find a scalar fitness function, or moreover, it does not exist.

On the other hand, existence of graph factors is an important theoretical problem which naturally appears in the study of complex networks, biological and social systems, etc [1][2][12][15][22]. For example, many communication and transportation systems can be represented as complex networks. A wide class of problem solutions is subject to finding connected subgraphs [9][20]. Other related works in this field include [23][24] and [25].

Apart from connective subgraphs, spanning subgraphs are also important in the study of communication in complex networks. One of the reasons is that malfunction of one or more nodes in a network will generally affect both global and local properties

of the remaining nodes. In this case, malfunction will cause some edges unusable and break the connectivity of the entire network.

A fractional factor is loosely defined as a spanning subgraph with prescribed node degrees. For example, a group of individuals interacting with each other in social networks combine into a global coupled network. A Hamilton cycle, or the longest cycle containing all the nodes of the network, is a spanning subgraph with each node adjacent to exact two edges of the graph  $G$ . Plummer and Saito [18], Nishimura [17], Ananchuen and Saito [3] obtained some properties for regular networks to be  $r$ -factor-critical in term of various concepts of closures, independence number or domination number, etc.

Although there are many theoretical results on the existence of graph factors, constructive way to find out them is more important in real applications. However, effective methods to find out factors still remain an open, complicated problem, and there are seldom any literature concerning about this up to now.

Inspired by these ideas, this paper focuses on the constructive study and effective searching of minimum fractional graph factors. In order to do so we propose an evolutionary model based on lattices for approximate factors. Up to now, we have not found this lattice based evolution and applications to graph factors. We also present a Markov analysis together with some experiments and discussions.

## 2 Problem Fractional Factors of Graphs

In this section, we will give detailed description of the problem. Let  $G=(V(G),E(G))$  be a simple and undirected graph, where  $V(G)$  and  $E(G)$ <sup>1</sup> denote the vertex set and the edge set of  $G$  respectively. For any  $S \subseteq V(G)$ , we use  $G-S$  to denote the subgraph generated by the vertices  $V(G) \setminus S$ . The symbol  $i(G-S)$  denotes the number of *isolated vertices* of  $G-S$ . A spanning subgraph is a subgraph that contains all the vertices of the original graph.

Let  $H$  be a spanning subgraph of  $G$ .  $H$  is called an  $r$ -factor of  $G$  if  $d_H(x) = r$  for each  $x \in V(G)$  where  $r$  is a nonnegative integer [6] and  $d_H(x)$  is the degree of a node. Fractional factors are natural generalization of factors. More precisely, let  $h(e) \in [0,1]$  be a function defined on  $E(G)$  and define

$$d_G^h(x) = \sum_{e \in E_x} h(e), E_x = \{xy : xy \in E(G)\} \quad (1)$$

Then  $d_G^h(x)$  is called the fractional degree of  $x$  in  $G$ . The function  $h(e)$  is called an indicator function.

**Definition 2.1** Let  $E^h = \{e : e \in E(G), h(e) \neq 0\}$  and  $G_h$  be a spanning subgraph of  $G$  such that  $E(G_h) = E^h$  and  $d_G^h(x) = r$  for  $x \in V(G)$ . Then  $G_h$  is said to be a fractional  $r$ -factor of the graph  $G$ .

One important fact is that the definition of fractional factors is subject to the indicator function. Among all the fractional  $r$ -factors, fractional 1-factors<sup>1</sup> attract special interests in the literature and they are also the main focus of this paper.

Suppose  $|V(G)| = N$  and the indicator function  $h(e) \in [0,1]$  for edge  $e$  adjacent to node  $i$  and  $j$ . We call  $h(e)$  the weight on the edge  $e$ . Let the symmetric weight matrix be defined as  $W = [w_{ij}]_{N \times N}$

Where  $w_{ij} = h(e)$  is the weight on the edge  $(i, j)$ . Sometimes complex networks also have weighted edges in the form of a matrix. Therefore an interesting problem is how to find fractional factors with a prescribed weight matrix.

We start with fractional 1-factors. First we suppose  $G$  has a fractional 1-factor  $F$ . In the following an edge will be denoted by  $e=(i, j)$  for simplicity. Then for all  $i, j = 1, 2, \dots, N$  we set define

$$x_{ij} = \begin{cases} 1, & e_{ij} \in F, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

**Table 2.1** Finding a minimum 1-factor in weighted network

**Input:**  $G$  with the edge weight matrix  $W = [w_{ij}]$

**Output:**  $F$ : a fractional 1-factor

Steps:

1. If  $rank(A) < rank(A, I)$ , then  $G$  has no fractional 1-factors. Otherwise if  $rank(W) = rank(W, I) \leq N$ , then  $G$  has fractional 1-factors.

2. Solve the linear equations (2.4). Let  $F$  be a fractional 1-factor with minimum edges among all the fractional 1-factors, then output  $F$ .

END

Consequently, for a network with given edge weight matrix  $W = [w_{ij}]_{N \times N}$  where  $w_{ij} \in [0,1]$ , the problem of finding a fractional 1-factor is equivalent to solving a group of linear equations as follows:

$$\begin{cases} \sum_{j=1}^N w_{ij} x_{ij} = 1, \\ x_{ij} \in \{0,1\}, \quad i=1,2,\dots,N \end{cases} \quad (3)$$

If written in matrix form, the group of equations is  $WX = I$ , where  $I$  is a matrix with diagonal members equals to 1, and  $W = [w_{ij}]_{N \times N}$ ,  $X = [x_{ij}]_{N \times N}$ . This implies that there are at most  $N$  equations with  $N \times N$  variables  $x_{ij}$  in  $WX = I$ . To solve the existence problem of fractional 1-factors we need to estimate the rank  $rank(W) = rank(W, I) \leq N$ .

Now we consider minimal fractional factors. For any network with given edge weight  $w_{ij}$ , finding a fractional 1-factor with minimum edges is a linear programming:

$$\begin{cases} \min \sum_{i,j} x_{ij} \\ \sum_{j=1}^N w_{ij} x_{ij} = 1, \\ x_{ij} \in \{0,1\}, \\ i=1,2,\dots,N. \end{cases} \quad (4)$$

There are at most  $N \times N$  equations with  $N \times N$  variables  $x_{ij}$  in (2.4). So it is straight forward to present an algorithm as in Table 2.1 for this problem.

## 3 Lattice Based Evolution

Evolution is an optimization scheme based on the mechanisms of natural selection with an important principle of survival of the fittest [11][13], etc. For a binary coding scheme with length  $L$ , all possible chromosomes form a space.

The notations  $V, E$  are often used when there is no confusion caused.

$$A = A_1 \times A_2 \times \dots \times A_L = \prod_{i=1}^L A_i \quad (5)$$

The system at time  $t$  is expressed by an architecture  $A(t)$  with environment variable  $I(t)$ . Evolution is a process  $\tau_t$ :

$$\tau_t : A(t) \times I(t) \times H(t) \rightarrow A(t+1) \quad (6)$$

where  $H(t)$  is a variable relating to history status of the system. Suppose  $(\mathfrak{R}, \langle \cdot \rangle)$  is a lattice. Then a lattice based fitness measure  $\mu_t$  is a mapping

$$\mu_t : A(t) \times H(t) \rightarrow \mathfrak{R} \quad (7)$$

In this setting, a lattice based evolution has the following formal description:

$$\begin{cases} \text{optimize } \mu_t(A(t)) \\ \tau_t : A(t) \times I(t) \times H(t) \rightarrow A(t+1) \\ H(t+1) = \tau_t(H(t), I(t)) \\ I(t+1) = \tau_t(I(t)) \end{cases}$$

Next we give some details about the proposed lattice measure. In the following we will use  $\mu : \Omega \rightarrow \mathfrak{R}$  to denote a general lattice measure where  $(\mathfrak{R}, \langle \cdot \rangle)$  is a linear lattice. A linear lattice is a lattice where the partial order is defined by a cone.

**Definition 3.2** Suppose  $\mathfrak{R}$  is a real linear space.  $C \subset \mathfrak{R}$  is a nonempty convex set satisfying

- (1) If  $x \in C$ , then  $\alpha x \in C$  for  $\alpha \geq 0$ .
- (2) If  $x \in C$ , then  $-x \in C$  for  $x = 0$ .

A partial ordering induced by a cone  $C$  means that  $x < y$  if  $y - x \in C$ . We often write  $x \leq y$  instead of  $x < y$  for simplicity. As usual, a measure will have some additive properties.

**Definition 3.3** Suppose  $B$  is a set consisting of some subsets of  $B \subset \Omega$  as its elements. We call  $B$  a  $\sigma$ -field if the following conditions are satisfied:

- (1)  $B \in B$ .
- (2)  $\bigcup_{j=1}^{\infty} E_j \in B$  provided  $E_j \in B$  for  $j = 1, 2, \dots$ .
- (3) If  $E \in B$  then  $\bar{E} \in B$  where  $\bar{E} = B \setminus E$ .

**Definition 3.4** Suppose  $B$  is a  $\sigma$ -field. A function  $\mu$  on  $B$  is called a lattice measure and  $(B, \mu)$  is called a lattice measure space if

- (1)  $\mu(E) \succ 0$  for any  $E \in B$ .
- (2)  $\mu(\emptyset) = 0$  provided  $\emptyset \in B$ .
- (3) If pairwise disjoint sets  $E_j \in B$  and  $\bigcup_{j=1}^{\infty} E_j \in B$  then  $\mu\left(\bigcup_{j=1}^{\infty} E_j\right) = \sum_{j=1}^{\infty} \mu(E_j)$ .

Now we present an example of lattice measure. Suppose  $C \subset R^d$  is the positive cone  $C = \{r = (r_1, \dots, r_d) : d_i \geq 0, i = 1, \dots, d\}$ . For a search space  $\Omega$  with a mapping  $f : \Omega \rightarrow C \subset R^d$ , let  $B = 2^\Omega$  and

$$\mu(B) = \sum_{x \in B} f(x), B \in \quad (8)$$

Then clearly the measure  $\mu$  is a lattice measure.

### 3.1 Dynamic Niche Scheme

Dynamic niche sharing model is proposed by B.L. Miller and Shaw M.J. in 1996 [16]. It attempts to identify peaks of the forming niches and uses dynamically identified peaks to classify all individuals as either belonging to one the dynamic niches, or else belonging to the non-peak category.

In the dynamic niche sharing scheme, the shared fitness value for an individual within a dynamic niche is its raw fitness value divided by the dynamic niche population size. Otherwise, the individual belongs to the non-peak category, and its niche count is calculated using the standard niche count. Applications have proved that dynamic niche sharing is more efficient than standard sharing techniques.

For the purpose of the graph factor problem in this paper, we present a variant dynamic model, the dynamic collaborative co-evolution model. The proposed model combines the advantages of dynamic niche sharing and co-evolution strategy. We define the model to be dynamic in the following sense. First, the population is dynamically divided into sub-populations. This dynamical process is controlled by the niche core. Secondly, the niches are dynamic which are determined by a niche core set consisting of present best solutions of the problem. Thirdly, the niche core set is dynamic in the sense that it is dependent upon the evolution process of all sub-populations.

Dynamic collaborative niche model is suitable for locating niches automatically. Since optimal solutions are not equally distributed in general, this dynamic model can effectively reduce computing time by eliminating those areas where there are no optimal solutions. By this technique we can find all possible optimal solutions in a relatively small computing complexity. Generally, there are two types of collaborations between sub-populations, the share collaboration via niche core, and the complete collaboration. In general, the first collaborative mode is direct and simple. However, in other problems, complete collaboration will be acceptable in that we can share good genes and parameters such as niche radii information, weights that have proved to be

effective in other niches. This can speed up the evolution process in whole.

In a dynamic collaborative niche model, the niche core represents niche centers of niches in all sub-populations. The core set contains optimal solutions of the co-evolution model finally. During evolution process, individuals in this set are possible optimal solutions. To initiate this set, we can choose some individuals randomly in the phenotype space, for instance.

Populations are dynamically divided into sub-populations via niche core. The dynamic scheme should be determined beforehand via certain measurement. For example, fitness value, distances between niches are possible measurements to determine the dynamic scheme. Each subpopulation will generate its niches during evolution which will be added to the niche core set.

For complete collaboration, one natural scheme is migration. In this scheme, evolutions are independent in all sub-populations via dynamic niche sharing. When new niches are generated, a portion of the best niches is selected in one subpopulation, and this portion migrates to other subpopulations by a predefined probability. In this mode, subpopulations will exchange good genes. Other schemes include mutual migrations, one-direction migration, infiltration, and so on. Figure 3.1 gives an illustration.

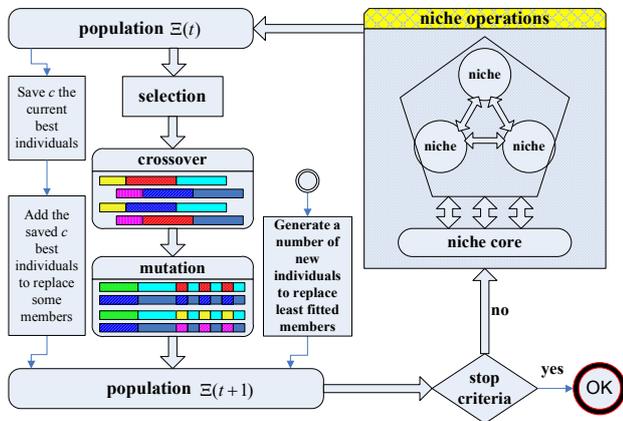


Fig. 3.1 An evolution flow chart.

• **Dynamic Fitness Sharing**

Suppose  $\Omega$  is the search space of binary strings with length  $L$  and  $(\mathfrak{R}, \langle \cdot \rangle)$  is a linear lattice. Denote the population by  $\varpi(t) = \{\Xi_i(t) : i = 1, \dots, \beta\}$  at the  $t$  generation with  $\beta$  number of  $\Xi_i(t)$ . Suppose each niche has a size  $\delta_i$ . Then the size of the whole population is  $n = \sum_{i=1}^{\beta} \delta_i$ . Let the raw fitness function be as follows

$$f : \Omega \rightarrow \mathfrak{R} \quad (9)$$

For  $x, y \in \Omega$  define the distance between them by  $d(x, y) =$  the Hamming distance of  $x, y$ . For one niche, the radius is the maximum distance of any two members. Suppose a member  $x$  currently belongs to a niche  $\Xi_i(t)$  with radius  $\sigma_i$ . Then define the niche share area of an individual  $x$  by

$$m(x) = \sum_{z \in \Xi} H(\sigma_i^{-1}d(x,z)) \quad (10)$$

Where  $H(t)$  is defined in (11) with  $h(t)$  a strictly increasing function defined in  $[0, 1]$ .

$$H(t) = \begin{cases} 1, & t \leq 0; \\ 1 - h(t), & t \in [0, 1]; \\ 0, & t \geq 1. \end{cases} \quad (11)$$

An example of the function  $h(t)$  is

$$h(t) = t^\alpha, \text{ with } \alpha > 0 \quad (12)$$

Now we write

$$dm(x) = \begin{cases} \frac{m(x)}{\delta_j}, & \text{if } x \text{ is within niche number } j; \\ 1, & \text{otherwise.} \end{cases} \quad (13)$$

Then the dynamic sharing fitness value of a member  $x$  is defined by

$$df(x) = \frac{f(x)}{dm(x)} \quad (14)$$

This is a shifted fitness value. If one particle is more like the existing optima, then its fitness value is reduced. Otherwise, it is left unchanged.

• **Cooperation and Co-evolution**

It is natural to view a Genetic Algorithm as a cooperative learner [8]. Clearwater et al. [8] define cooperation as follows: "Cooperation involves a collection of agents that interact by communicating information to each other while solving a problem." Clearly, by viewing the population members of a GA as agents, and the crossover operation as information exchange, the GA can be considered to be a cooperative system.

Inspired by this approach, we present a parallel cooperative learning model (PCLM). The cooperative system is called co-evolution. We divide the whole population into several sub-populations which will evolve in parallel. An mechanism is assigned to each sub-population which can be implemented by an agent. These agents act as CPUs in traditional parallel architecture and form one of the two models: the master-slave model and the island model.

In the master-slave model, one special agent is defined as master agent. Each sub-population is assigned a slave agent. The master agent supervise the whole population and assign whole tasks and opera-

tions. The master agent chooses good individuals (seeds) from the whole population and migrate the seeds to all sub-populations. Agents of the subpopulations received seeds and perform sorting, selection and discarding. Migration operations can be one-to-more, or one-to-one (Figure 3.2).

The second model is the island model. In this model, one agent is assigned to each sub-population. Agent supervises the evolution process of its sub-population. Sub-populations exchange good individuals at a certain time interval mutually. Migration operations can be one-to-more, or one-to-one (Figure 3.2).

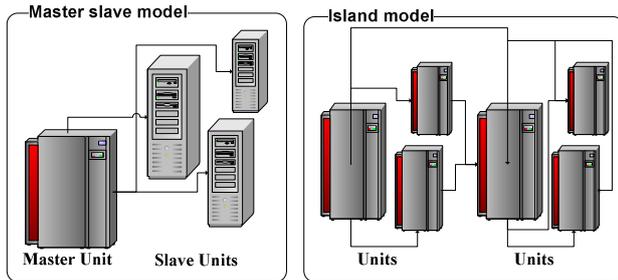


Fig. 3.2 Master-slave and island model.

The role of migration operators is to exchange good individuals between sub-populations. For island model, there are two migration schemes, the one-to-more and one-to-one scheme.

In the one-to-more scheme, each sub-population shares the current best individual with all other sub-populations in every generation. Whenever it receives one good individual, the worst is discarded according to fitness value. In the one-to-more scheme, the special one is often selected as the global best individual. The one-to-more migration will speed up convergence and reduce population diversity.

The second migration is one-to-one. Diversity is one of the main concerns in this scheme. In this scheme, each subpopulation defines a send rate and sends probability and send best individuals according to the rate. The rate determines how many individuals are selected to send while the probability determined how often this operation occurs. The selection of sent members can be random, or fitness preferential. The destination of sending can be totally mutual, neighbour wise, or ring topology. In the neighbourhood topology, one can define neighbours by geometric or concept hierarchy (Figure 3.3).

#### 4 Evolution Scheme for Minimal Factors

In this section, we will present an evolution scheme for the searching of minimal 1-factors for weighted graphs. From previous discussions we know that multi-population and coevolution schemes will be

used to guarantee better evolution performance. However, there are still some problems concerning the coding and genetic operations. Now we will focus on these technical details.

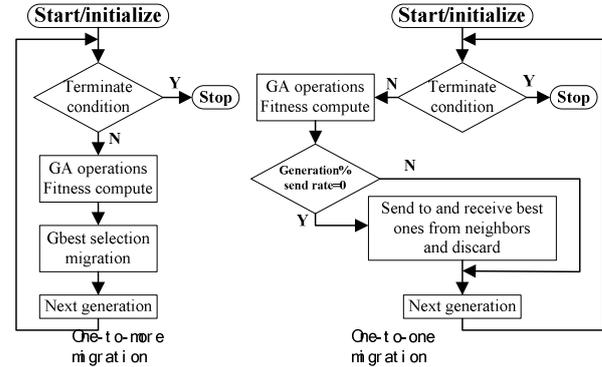


Fig. 3.3 One to one and one to more migration.

In order to do so, we need to design an encoding scheme and fitness evaluation method. First we start with the encoding scheme. Let  $\chi$  be the set of symmetric matrices as in Section 2 consisting of 1 and 0 as its elements. Then for  $X \in \chi$  we define the genotype of X straight by

$$x_{11}, \dots, x_{1N}, \dots, x_{N1}, \dots, x_{NN} \quad (15)$$

Here we use redundant encoding scheme in order to simplify genetic operations in the following. Next we explain genetic operations. First we start with the one-point crossover scheme. Multi-point crossover can be described similarly. Choose randomly two cross positions denoted by  $i, j$ . Then the crossover algorithm is described as follows

#### Crossover algorithm

```
for(int m=i;m<N;m++) {
    for(int n=i;n<N;n++)
    {
        if((m==i)&&(n<j)) continue;
        if((m<j)&&(n==i)) continue;
        //swap
        swap(x[m,n], y[m,n]);
    }
}
```

According to the traditional definition of mutation, in order to keep the symmetry of the matrix, we need to mutate a pair of positions simultaneously. Therefore these two operations are illustrated in Figure 4.1.

#### 4.1 Cooperation and Co-evolution

Now we define the fitness function of the 1-factor problem. If we use  $\delta_{ij}$  as the Kronecker symbol, that is,  $\delta_{ij} = 1$  if  $i = j$  and zero else, and let  $\Theta = [\delta_{ij}]_{N \times N}$ . For a matrix  $A = [a_{ij}]$ , we define its absolute trace to be

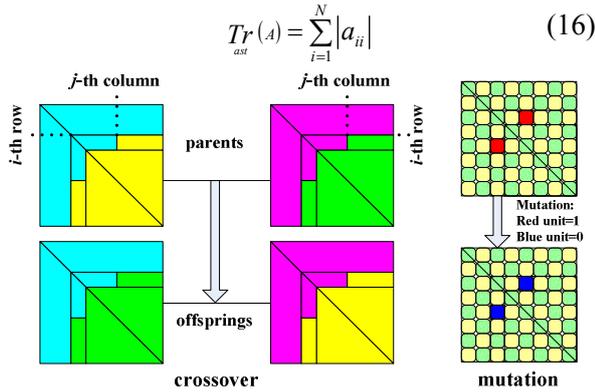


Fig. 4.1 Crossover and mutation operations.

Then the degree of a connection matrix towards forming a 1-factor can be measured by

$$f_1(X) = \sum_{i=1}^N \left| \sum_{k=1}^N w_{ik} x_{ki} - \delta_i \right| = T_{ast}(WX - \Theta) \quad (17)$$

By the definition of factors, isolated nodes are prohibited. Therefore, we need a restriction function to avoid isolated node. Suppose the minimal node of the graph is defined by  $d(X)$ .

$$d(X) = \min_{1 \leq i \leq N} \sum_{j=1}^N x_{ij} \quad (18)$$

Then one possible restriction function is as follows:

$$p(X) \geq \varepsilon, \text{ where } \varepsilon > 0 \quad (19)$$

In the formula  $\varepsilon$  is a parameter to control the degree of restriction. Next we consider the problem of minimal factors. By definition, a minimal factor is a subgraph which has least number of edges. Thus this can be measured by

$$f_2(X) = \sum_{i=1}^N \sum_{j=1}^N x_{ij}^2 \quad (20)$$

Now we define the fitness function as follows

$$f: \mathcal{X} \rightarrow R^2, f(X) = (f_1(X), f_2(X)) \quad (21)$$

Then the natural partial ordering in  $R^2$  can be defined by 1 the positive cone:

$$C = \{(x, y) \in R^2 : x > 0, \text{ or } x = 0 \ \& \ y > 0\} \quad (22)$$

Therefore the searching of minimal factors is an optimization with restriction

$$\begin{cases} \text{minimize } f(X) \text{ w.r.t } \prec \\ p(X) \geq \varepsilon \end{cases} \quad (23)$$

Let  $\Upsilon = f(\mathcal{X})$ . Then clearly  $\Upsilon$  under the partial order  $\prec$  is a lattice.

## 4.2 Lattice Based Evolution

Suppose  $\Omega$  is the search space and  $(\mathfrak{R}, \prec)$  is a lattice. A fitness function is defined on  $\Omega$  by  $f: \Omega \rightarrow \mathfrak{R}$ . We assume that the image set  $\mathfrak{R} = f(\Omega)$ , that is, the map  $f$  is a surjection. If this is not satisfied, then we need

to show that the image set  $f(\Omega)$  is a sub-lattice in advance. Therefore, for each  $y \in \mathfrak{R}$ , the set  $f^{-1}(y)$  is not empty. In the following, we will use  $f^{-1}(y)$  to denote one of its members  $x$  and hence  $f(x) = y$ .

In order to describe evolution on the lattice  $\mathfrak{R}$ , we need first to define the selection operation which is different with traditional scalar valued fitness functions. Now we assume that the population is  $\Xi = \{x_1, \dots, x_n\}$  with corresponding fitness values  $y_i = f(x_i)$ . By applying the elitist selection scheme, the current best individual is defined by

$$x = f^{-1}(y), \text{ where } y = y_1 \wedge \dots \wedge y_n \quad (24)$$

Notice that this best individual is possibly a new one which is not in the current population. This member is formally denoted by the function  $x = \min\{x_1, \dots, x_n\}$ . Based on this definition, we propose a sorting algorithm on lattices.

### Sorting algorithm on lattices

1. Input sequence  $\Xi = \{x_1, \dots, x_n\}$ .
2. Output sorted sequence  $\Theta = \text{sort}(\Xi) = \{z_1, \dots, z_n\}$ .
3.  $z_1 = \min\{x_1, \dots, x_n\}$ . Set  $i = 1$  and  $P_1 = \{x_1, \dots, x_n\}$ .
4. Begin loop.
5.  $z_i = \min P_i$ .
6. If  $z_i \in P_i$ , then let  $P_{i+1} = P_i \setminus z_i$ . Let  $i = i + 1$  and goto Step 4.
7. If  $z_i \notin P_i$ , then let  $i = i + 1$  and goto Step 5.
8. End loop.

It is clear that the sorted sequence is totally ordered, that is

$$f(z_1) \prec f(z_2) \prec \dots \prec f(z_n) \quad (25)$$

The proposed sorting operation here is variation of normal sorting in that the resulting sequence may be different with the original one. Therefore we can propose a lattice based genetic algorithm as follows.

### Lattice based genetic algorithm

1. Initialize population  $\Xi(0) = \{x_1, \dots, x_n\}$ .
2. Begin loop. Set  $t = 1$ .
3. Compute fitness values of the current population. Perform lattice based sorting.
4. Check terminal condition. If check result is yes then go to Step 7. Otherwise go to Step 5.
5. Perform genetic operations. Do selection, crossover, mutation and lattice sorting successively.
6. Check restriction condition. If check result fails, then replace the failed individual by random generated new individuals.

7.  $\Xi(t) \Rightarrow \Xi(t+1)$ ,  $t = t + 1$ . Go to Step 3.
8. End loop.

### 5 Markov Chain Analysis

In this section, we will present a simple convergence analysis of the lattice based evolution by Markov chains. Suppose  $\Omega$  is the search space and  $(\mathfrak{R}, <)$  is a lattice. Denote the population by  $\varpi(t) = \{\Xi_i(t) : i = 1, \dots, \beta\}$  the population set at time  $t$  with a fixed number of subpopulations  $\Xi_i(t)$ . Suppose each subpopulation has a fixed size  $\delta$ . Then the size of the whole population is  $n = \beta\delta$ .

First we define a scalar energy function in order to switch the minimal optimization to maximal. Therefore if an individual  $X$  is more fitted, it has greater energy.

$$F(X) = \frac{1}{1 + f_1(X) + f_2(X)} \quad (26)$$

Notice that the niching operations are needed for the searching of multiple solutions. When we analyze the convergence behaviour of the evolution one solution should be specified. Therefore we assume that there is no niching operations. Hence the population  $\varpi(t)$  at time  $t$  is dependent only on the population at time  $t-1$  and hence it is a Markov chain.

Let  $L = N \times N$  be the string length of the genotype where  $N$  is the number of nodes of the graph. The size of the search space is  $m = |\Omega| = 2^L$  and the set of possible populations is  $\Omega^n$  with the number of different populations being

$$M_T = \binom{C_{m+\delta+1}^{m-1} + \beta - 1}{C_{m+\delta+1}^{m-1} - 1} \quad (27)$$

Each individual in the search space can be considered as an integer with binary bits just as its genotype string. We will use this integer to represent a string or an individual later. For a specific population, we can take it as one string with length  $nL$ . Define  $i^*(\varpi)$  to be the best individual defined by (21) in the population  $\varpi$ .

By Section 4 we know that each population generated by evolution is totally ordered. In fact, their best individuals form a totally ordered set. Therefore, we only consider those populations with their members totally ordered in ascending order. Denote this set of populations by  $A$  and let  $M = |A|$ . Therefore

$$\{i^*(\varpi) : \varpi \in A\} \text{ is totally ordered} \quad (28)$$

Let  $Z(i, \varpi)$  be the occurrences of the individual  $i$

in the whole population  $\varpi$ , and  $Z_j(i, \varpi)$  be the occurrences of the individual  $i$  in the subpopulation  $\Xi_j$ . Then

$$\begin{cases} \sum_{i=0}^{m-1} Z(i, \varpi) = n, \sum_{i=0}^{m-1} Z_j(i, \varpi) = \delta \\ \sum_{j=1}^{\beta} Z_j(i, \varpi) = Z(i, \varpi) \end{cases} \quad (29)$$

For genetic operators, the following selection probability will be applied for subpopulation  $\Xi_j$  and  $\varpi$ .

$$r_j(i, \varpi) = \frac{F(i)Z_j(i, \varpi)}{\sum_{r=1}^m F(r)Z_j(r, \varpi)} \quad (30)$$

$$r(i, \varpi) = \frac{F(i)Z(i, \varpi)}{\sum_{r=1}^m F(r)Z(r, \varpi)} \quad (31)$$

We will use  $Q = [Q_{ij}]_{M \times M}$  to denote the transition matrix of the Markov chain. Suppose the initial distribution vector of the  $M$  possible populations is

$$q^{(0)} = \{q_1^{(0)}, q_2^{(0)}, \dots, q_M^{(0)}\} \quad (32)$$

Then after  $t$  generations the distribution is

$$q^{(t)} = \{q_1^{(t)}, q_2^{(t)}, \dots, q_M^{(t)}\} = q^{(0)}Q^t \quad (33)$$

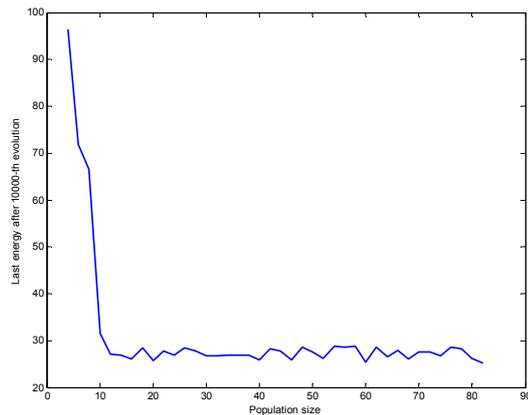
Suppose the members of the matrix  $Q^t$  are denoted by  $Q^t = [Q_{ij}^{(t)}]_{M \times M}$ . Convergence problems concern in fact the limit distribution  $q^{(\infty)}$ . Define  $P(j, \varpi)$  to be the probability that individual  $j$  occurs in the population  $\varpi$ . Next in order to simplify notations we will ignore the effect of subpopulations and suppose  $c = 1$  (which is called the modified elitist strategy). Notice that the best individual itself permits genetic operations, and hence it is clear that ([21][13]) the transition probability is

$$Q_{\mu, \beta} = \begin{cases} \frac{P(j, \mu)^{Z(j, B)}}{n! \prod_j Z(j, B)!}, i^*(\mu) \in B \\ 0, \text{ otherwise} \end{cases} \quad (34)$$

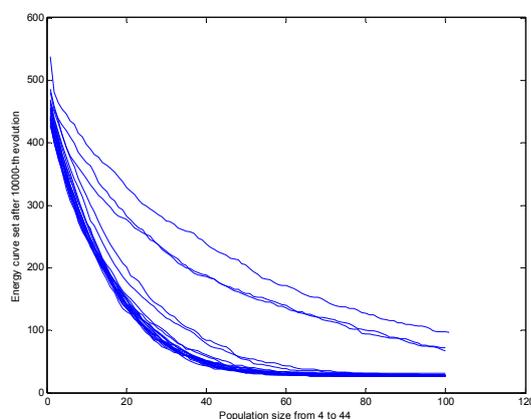
Similar to [21][19] the transition matrix  $Q = (Q_{\mu, B})$  from a population  $\mu$  to a population  $B$  has  $J$  sub-matrices  $Q(i)$  of size  $N(i) \times N(i)$  for  $i = 1, \dots, J$  as the diagonal elements as follows (some components to the lower left of the diagonal matrices can have non-zero values)



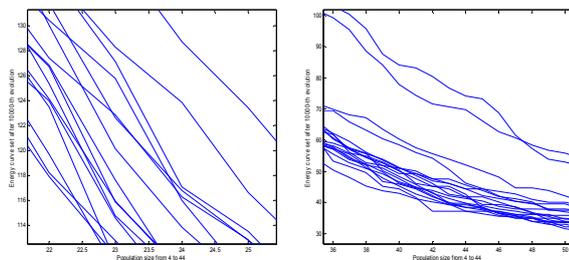




**Fig. 6.5** Last energy comparison of evolution with population size from 4 to 82.



**Fig. 6.6** Energy curves of evolution with population size from 4 to 15.



**Fig. 6.7** Local energy curve view with varying population.

## 7 Conclusion

In this paper we propose an evolutionary technique to find minimal 1-factors for graphs. Although we have not found similar approach that combines evolutionary technique with graph factors, it shows apart from theoretical existence results, the new computing technique can provide a new powerful roadway to find out these factors. We proposed a lattice based genetic algorithm in which the fitness function is not scalar and presents a simple Markov analysis.

Finally we would point out that the theoretical analysis by Markov chains of latticed based GA has

a long way to go. Also, experiments show that the convergence of GA in searching factors is slow and hence technique to speed up the convergence is necessary before the proposed algorithm can be used in real-time networks.

## Acknowledgements

This project is carried out under the "Taishan Scholar" project of Shandong China.

Research is also supported by the Natural Science Foundation of China (No.60873058, No.60743010, No.60673047), the Natural Science Foundation of Shandong Province (No. Z2007G03), and the Science and Technology Project of Shandong Education Bureau.

## References

- [1] Adamic L. (1999) The small worldWeb. Proceedings of the European Conf. on Digital Libraries: 443-452
- [2] Albert R., A-L. Barab'asi (2002) Statistical mechanics of complex networks. Reviews of modern physics 74:47-97
- [3] Ananchuen N., Saito A. (2003) Factor criticality and complete closure graphs. Discrete Mathematics 265:13-21
- [4] Anstee R.P. (1990) Simplified existence theorems for (g; f)- factors. Discrete Applied Math 27:29-38
- [5] Bergh F. van den and Engelbrecht A. P. (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evolutionary Computation 8(3):225-239
- [6] Bollob'as, B. (1998) Modern Graph Theory. Springer-Verlag, New York
- [7] Brown J.R. (1974) Shortest Alternating Path Algorithm Networks 4(4):311-334
- [8] Clearwater S.H., Hogg T., and Huberman B.A. (1992) Cooperative problem solving. in Computation: The Micro and Macro View, Singapore: World Scientific: 33-70
- [9] Dorogovtsev S.N., Mendes J.F.F. (2003) Evolution of Networks. Oxford University Press, Oxford
- [10] Edward R. Schinerman and Ullman, D.H. (1997) Fractional Graph theory. John Wiley and Son Inc., New York
- [11] Goldberg D.E. (1989) Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading, MA
- [12] Kleinberg, J. (2000) The small-world phenomenon: an algorithmic perspective. in Proceeding of the 32nd annual ACM symposium on theory of computing, New York: 163-170
- [13] Li M.Q., Kou J.S., Lin D., and Li S.Q. (2002) Fundamental Theories and Applications of Genetic Algorithms. Science Press, Beijing

- [14] Ma Y.H. and Liu G.Z. (2003) Isolated toughness and existence of fractional factors in graphs. *Acta Appl. Math. Sinica* (in Chinese) 26:133-140
- [15] Milgram Stanley (1967) The small world problem. *Psychology Today* 1(1):60-67
- [16] Miller B.L. and Shaw M.J. (1996) Genetic algorithms with dynamic niche sharing for multimodal function optimization. in *Proc of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, IEEE:786-791
- [17] Nishimura T. (2002) A closure concept in factor-critical graphs. *Discrete Mathematics* 259:319-324
- [18] Plummer M. and Saito A. (2000) Closure and factor-critical graphs. *Discrete Mathematics* 215:171-179
- [19] Rudolph Gunter (1994) Convergence analysis of canonical genetic algorithms. *IEEE Trans Neural Networks*, special issue on Evolutionary Computing 5(1):96-101
- [20] Strogatz Steven H. (2001) Exploring complex networks. *Nature* 401(6825):268-276
- [21] Suzuki J. (1995) A Markov chain analysis on simple genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics* 25(4):655-659
- [22] Watts D.J. (2004) The new science of networks. *Annu Rev Sociol* 30:243-270
- [23] Sule Gunduz Oguducu, A. Sima Uyar.(2004) A graph based clustering method using a hybrid evolutionary algorithm. *WSEAS Trans. on Mathematics* 3(3):731-735.
- [24] Yaakov Hacoheh-Kerner, Yaakov Intrator, Vadim E. Levit (2004) A  $O(n^2)$  heuristic algorithm for the maximum matching problem in bipartite graphs. *WSEAS Trans. on Mathematics* 3(3):521-526.
- [25] Qiang Hua, Bin Wu, Hao Tian (2008) A detecting peak's number technique for multimodal function optimization, *WSEAS Trans on Information Science and Applications*, 2(5): 37-43.