

Parallel Island-based Estimation of Distribution Algorithms for Wireless Network Planning

Feng Liu¹, Yonghua Zeng², Yousuo Zou², Juan Liu¹, Huaibei Zhou^{1,2,*}

¹Computer School of Wuhan University, Wuhan, P.R. China

² State Key Lab of Software Engineering, Wuhan University, Wuhan, P.R.China

* Corresponding author, E-mail: zhouhuaibei@vip.sina.com

Abstract--- Wireless network planning is an important problem to the communication companies. To solve this problem, many models were proposed, which overlooked different profits on the same acreage in different regions. Genetic algorithms (GAs) and similar algorithms were put forward by many papers based on those models. However, those models cannot reflect the companies' aims that make maximum profit with minimum cost. In this paper, we put forward a novel model with profit weight and penalty factor and design a new simulation experiment about the model. In order to gain the optimal solution quickly, a kind of novel algorithms--parallel island-based estimation of distribution algorithms (PEDA)--is proposed based on [5]. Through simulation tests, we find that our algorithms outperform other stochastic heuristic search algorithms.

Index Terms---Estimation of Distribution Algorithm (EDA), PEDA, Genetic algorithms, Wireless Network Planning

I. INTRODUCTION

The selection of a good set of sites among the candidate base transceiver stations (BTSs) is a critical problem that the telecommunication companies must solve when building a wireless network. Many solutions were proposed based on the hypothesis that the problem comes down to serving a maximum surface of a geographical area with a minimum number of BTSs [1, 2, 4]. According to the hypothesis, the unicast set covering problem (USCP) was proposed in [2]. In the model, genetic algorithms and parallel island-based genetic algorithms were introduced [2, 4].

The USCP model was proposed by the assumption that company's profit is linear proportion to the covered acreage. However, the companies make different profits on the same acreage in different regions. For example, the benefits they get in the urban are more than that in the suburb given the same covered acreage. Furthermore, the uncovered region can negatively affect the profit of another covered region. That is to say, the assumption cannot reflect the companies' aims that making maximum profit with minimum cost.

A reasonable model should take account of profit weight in different covered area and the loss weight (penalty factor) of the uncovered area. In this paper, we proposed a profit-based covering model (PBCM) which employs profit weight and penalty factor, and put forward a kind of novel algorithms, parallel island-based estimation of distribution algorithms (PEDA), to solve the problem.

II. PROBLEM MODELING

In [2], the covered area of a transmitter (called a cell) can be discretized by grids. Thus the cells of BTSs can be represented by a finite set of grid as figure 1 shows.

Then the PBCM model can be described as follows:

Let $I=\{v_i\}$ be the set of all discretized regions, $L=\{l_i\}$ be the set of potentially covered regions, $\bar{L}=I-L$ be the set of uncovered regions and $M=\{m_i\}$ be the set of all candidate base locations. The set $E=\{e_{ij}\}$ is the set of edges e_{ij} , which denotes the j^{th} region is covered by i^{th} transmitter location. Then the graph $G=\{M \cup L, E\}$ is defined. Let $L'=\text{Neighbors}(M', E)$ denote the covered regions of all transmitter locations in M' , where $M' \subseteq M$ and $\text{Neighbors}(M', E)=\{u \in L \mid \exists v \in M', (v, u) \in E\}$. Let $|M'|$ denote the number of element in M' . Let $W[i]$ denote relative profit of the i^{th} covered region, that is to say, the weight of the i^{th} covered region. Let $P[k]$ denote relative loss if the k^{th} region isn't covered, where $p[k]$ is always minus. Therefore, the profit of the candidate base locations in M' is following:

$$\text{profit}(M') = \begin{cases} \sum_i W[i] + \sum_j P[j] & \text{if } (\sum_i W[i] + \sum_j P[j] \geq 0) \\ 0 & \text{other} \end{cases} \quad (1)$$

here $i \in L'$ and $j \in \bar{L}$ $L' = \text{Neighbor}(M', E)$

Thus the problem is becoming how to search a minimum subset of candidates that can get maximum profit, in other words, how to find a subset $M' \subseteq M$ that $\text{profit}(M')$ is maximum and $|M'|$ is minimum.

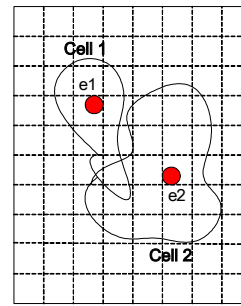


Figure 1. Two potential transmitters e1 & e2, and associated cell1 & cell2

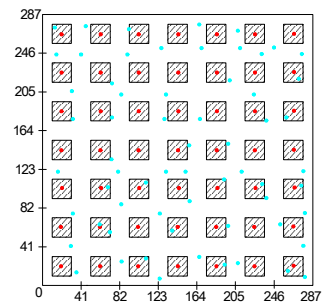


Figure 2. “●” denotes primary transmitters location, “●” denotes randomly selected candidate, “■” represents the superprofit region

However, the problem is known to be NP-hard[2,4]. Since the goal of maximizing the $profit(M')$ and minimizing the $|M'|$ is twofold, we construct a new objective function to combine the two factors as follows:

$$f(M') = \frac{profitRate^\alpha}{|M'|} \quad (2)$$

where $profitRate = 100 \times profit(M')/profit(M)$, the percent of M' 's profit in that of all candidate set M . The parameter α can be tuned and $\alpha = 2$ in [2].

Based on [2], we put forward a novel simulation experiment to solve the PBCM problem mentioned above.

As figure 2 shows, 287×287 point grids represent an open-air flat discretized area to cover. Suppose each transmitter can cover 41×41 grids, and then the total coverage is obtained if 49 primary transmitters were distributed regularly, forming a 7×7 grid structure. In order to simulate the profit and loss, we define $w[i]$ the profit weight if the i^{th} discrete region is covered. Suppose 49 superprofit regions were distributed regularly and each region consists of 21×21 grids. Let $w[i]$ be 1.1 if the i^{th} covered region is among superprofit regions, otherwise $w[i]$ be 1.0. Similarly, we define $P[k]$ the penalty factor if the k^{th} region wasn't covered. In order to simplify the implementation, $p[k]$ is set to -0.1. It is obvious that the $profitRate$ is 100 if and only if 49 primary transmitters are distributed regularly. Then the candidate set M is composed of 49 primary transmitter locations and randomly selected C ones. The $49+C$ candidates are shuffled randomly. Each candidate can be chosen or not, so the total searching space is $2^{(49+C)}$. The best solution is $f(M') = 100^2 / 49 = 204.08$.

III. METHODS AND ALGORITHMS

Parallel genetic algorithms (PGAs) are employed in [2,4] to solve the USCP problem and the algorithms show high accuracy and robustness. [5,6] proposed an estimation of distribution algorithm (EDA) as a novel evolution algorithm. In the light of these ideas, we propose parallel estimation of distribution algorithms (PEDA) to solve the PBCM problem.

A. Estimation of Distribution Algorithms (EDA)

Genetic Algorithm is one of stochastic heuristic search methods, which store more than one solution each iteration and converge more quickly. So it is widely used in searching problems as an optimization algorithm. However, the researcher requires experiences in order to choose the suitable values for the parameters in the algorithm. Therefore, a new type of algorithms--Estimation of Distribution Algorithms (EDAs)-- was introduced [5,6]. The algorithms try to make easier to predict the movements of the populations in the search space as well as to avoid the need for so many parameters. Like GAs, the algorithms are also based on the populations and have a theoretical foundation on probability theory.

Unlike GAs, the new individuals in the next population are generated without crossover or mutation operators in EDAs. In fact, the new individuals are randomly reproduced by a probability distribution estimated from the selected individuals in the previous generation. At the same time, in EDAs the interrelations between the different variables representing the individuals are expressed clearly by means of the joint probability distribution associated with the selected individuals at each generation.

Suppose a population of R chromosomes (individuals) and each chromosome composed of n genes like X_1, X_2, \dots, X_n . Then a generic schematic of EDA approaches are shown as figure 3 and the essential steps are following:

Step1. Generate randomly R individuals, composed of n -dimension, in D_0 generation.

Step2. Select m ($m < R$) best individuals, denoted by D_{h-1}^m , from the population in $h-1$ generation following a criterion.

Step3. Induce the n -dimensional probabilistic model that better represents the interdependences between the n variables. The model can be presented by a directed acyclic graph (DAG). This is the most crucial step in EDA.

Step4. Propagate R new individuals, which constitute the new population D_h , by carrying out the simulation of the probability distribution.

Steps 2, 3 and 4 are looped until the terminating condition is satisfied.

Many ways to estimate the joint probability distribution associated with the selected individuals from the previous generation in discrete domains and more details are introduced in [6].

In this paper, we suppose that the n -dimensional joint probability distribution factorizes like a product of n univariate and independent probability distribution. So the famous algorithm UMDA (univariate marginal distribution algorithm) is adopted to form the directed acyclic graph (DAG) and propagate the next generation in our program.

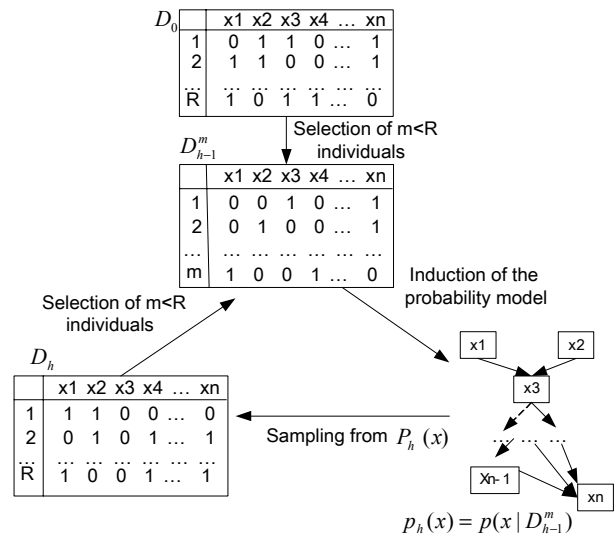


Figure 3. Illustration of EDA approaches in the optimization process

B. Parallel Estimation of Distribution Algorithms (PEDA)

Based on the island-based model, we proposed a parallel estimation of distribution algorithm (PEDA). Like algorithms in [1, 2, 4], the main steps of PEDA are following: firstly, the initial population is divided into a fixed number of subpopulations, called islands or demes, and each subpopulation evolves independently using EDA simultaneously. Secondly, in order to let some islands benefit from the information found by others, some individuals are allowed to migrate from one island to another after evolving a fixed number generations independently. This procedure is called migration.

The methods of migration between subpopulations are called connection topology. As Figure 4 shows, one-direction ring (ODR), dual-direction ring (DDR) and multi-direction ring (MDR) are three common types of connection topology [7].

Migration strategy is crucial to parallel island-based evolutionary algorithm. According to [1, 2], we adopt the best/random strategy that a random individual in a subpopulation is replaced by the best one in another subpopulation. Migration epoch, which is the number of generations evolving independently in subpopulations each migration, is vital to convergence. Here, we adopt the strategy that every 4 generations take a migration.

C. Implementation of Algorithm

To solve the PBCM problem above, PEDA is adopted in this paper. The algorithm is described as follows:

Input: size of the population (*popSize*), number of subpopulation (*subNum*), probability of selection (*Pselect*), initialization probability (*r*), migration epoch (*epo*)
Output: global optimal solution (the optimal BTSs)
Procedure:
Step 1: initialize a population with *popSize* individuals, among which each gene is set to 0 under probability *r* independently
Step 2: divide the population into a *subNum* subpopulations averagely and compute the size of subpopulation (*subPopSize*=*popSize*/*subNum*)
Step 3: select a better subpopulation for each one, which consists of *subPopSize***Pselect* better individuals.
Step 4: induce a DAG graph using UMDA
Step 5: propagate *subPopSize* individuals as next subpopulation by carrying out the simulation of the probability distribution
Step 6: conduct migration after each subpopulation evolves *epo* generations independently
Step 7: repeat step 3, 4, 5, 6 until the termination condition is satisfied

Consistent with encoding in [1, 2, 4], an individual is composed of $(49+C)$ -length binary string, with each bit (each gene) corresponding to a candidate location. The location is selected if the corresponding gene is 1, otherwise it is not selected. For each gene, the probability of being 0 is *r* when gene is initialized. Formula (2) defined in section 2 serves as the fitness function to discriminate individuals. The algorithms are implemented using Java (j2sdk1.4.2)

IV. RESULT AND DISCUSSION

In this paper, PBCM model and simulation method discussed in section 2 are used to simulate wireless network planning. In the test, the *C* value is 111, and thus the size of solution space is $2^{160}=1.461 \times 10^{48}$. Firstly, let *Popsiz* be 1280, *subNum* be 8, *r* be 0.5 and *Pselect* be 0.7 and then we discuss the convergence with parameter variations. The 5 tests are implemented on fixed parameters and results are shown as table 1.

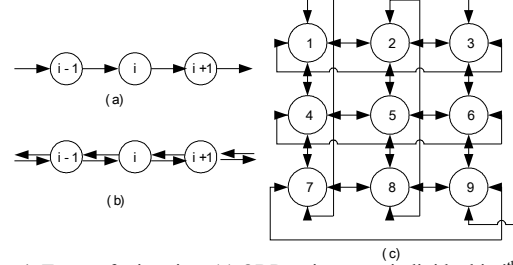


Figure 4. Types of migration. (a) ODR: migrate an individual in i^{th} subpopulation to $(i+1)^{th}$; (b) DDR: migrate an individual in i^{th} subpopulation to $(i+1)^{th}$ and $(i-1)^{th}$; (c) MDR: migrate an individual in i^{th} subpopulation to four neighbors.

TABLE I
CONVERGENCE GENERATION AND VALUE OF DIFFERENT METHODS
(a) Convergence generations with different migration using GA^a

	NM ^b	ODR ^c	DDR ^c	MDR ^c
1 st test	727	63	69	920
2 nd test	406	165	326	360
3 rd test	302	99	100	99
4 th test	367	103	250	58
5 th test	296	94	69	251

(b) Convergence generations with different migration using EDA

	NM ^b	ODR ^c	DDR ^c	MDR ^c
1 st test	54	144	72	61
2 nd test	55	74	63	57
3 rd test	50	178	70	65
4 th test	51	135	77	96
5 th test	49	111	67	70

(c) Convergence values with different migration using GA^a

	NM ^b	ODR ^c	DDR ^c	MDR ^c
1 st test	167.73	194.4	196	195.6
2 nd test	178.3	198.2	198.62	192.1
3 rd test	171.1	199.5	203.44	194.2
4 th test	167.1	200.23	198.54	202.8
5 th test	174.83	189.36	190.4	189.7

(d) Convergence values with different migration using EDA

	NM ^b	ODR ^c	DDR ^c	MDR ^c
1 st test	174.65	198.8	201.36	200.54
2 nd test	170.33	198.2	200.71	200.76
3 rd test	171.72	200.76	196.41	201.6
4 th test	177.25	198.77	199.12	199.72
5 th test	182.84	195.06	195.69	194.84

Note: ^a: Uniform crossover is used as well as in [4], in which crossover rate is 0.7, mutation rate is 0.1 and other parameters are as same as ones in EDA. ^b: NM, short for Non-migration, doesn't adopt island-based parallel model. ^c: uses the island-based parallel model.

Table (a) and (c) show that the convergence time of standard GAs without migration is long. The best convergence generation is 302 and worst 727. Meanwhile, the convergence values are bad, between 167.7 and 174.8. The convergence efficiency has been improved much when island-based model is adopted, with the best convergence generation 58, best convergence value 203.44 and worst convergence value 192.1. This is consistent with [2,4]. The difference of convergence efficiency is insignificant when employing different migration methods. Moreover, we find that convergence efficiency and values are not stable when using standard GAs. For instance, the best convergence generation is 58, but the worst is 920 using MDR migration strategy. For convergence values, the best is 203.44, but worst is 192.1. On the other hand, table (b) and (d) show that convergence is very quick in EDA without migration. Generally, the convergence generation is about 50. The convergence values have been improved much compared to standard GAs. When PEDAs is adopted, the convergence generation is stable for five tests regardless of migration strategies. The average convergence generation is around 100 and the convergence values are almost above 198. On the whole, PEDAs performs better than standard EDA and PGAs on convergence efficiency. In PEDAs, MDR strategy is the best among all migration strategies; as table (d) shows global optimal solution 204.8 can be gained three times in five tests.

In order to see the effect of parameter r on convergence result in PEDAs, we examine the convergence generation and values with different r (see Table 2).

TABLE 2

VALUE AND GENERATIONS OF CONVERGENCE WITH r^a

r	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CV ²	204.08	204.08	204.08	204.08	204.08	204.08	204.08
CG ³	54	51	53	52	65	64	82

^a: $popSize$ is 1280, $subNum$ is 8, and migration strategy is MDR; ²: CV refers to convergence values; ³: CG refers to convergence generation

From Table 2, we can observe that global optimal solution can be gained for different r , which has little effect on convergence efficiency.

To discern the impact of parameter $popSize$ on convergence efficiency, the changes of fitness (see Figure 5) and CPU executing time (see Table 3) are observed according to different $popSize$ with $subNum=8$.

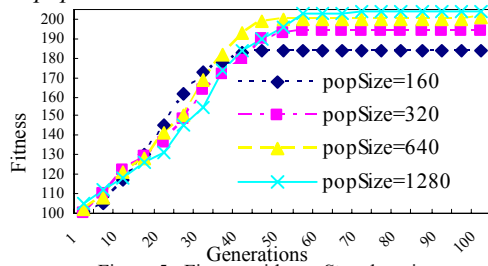


Figure 5 Fitness with $popSize$ changing

TABLE 3

CPU EXECUTING TIME WITH DIFFERENT $popSize$

$popSize$	160	320	640	1280
CPU time (s)*	53.01	98.92	345.56	441.92

* running environment: Xeon 2.7GHz, memory 1G DDR 400 Linux 2.4.21

As Fig 5 shows, convergence values increase along with the increase of $popSize$ given $subNum=8$. When $popSize$ reaches 1280, global optimal solution can be gained three times in five tests within 100 generations. CPU executing time is linear with $popSize$. However, the CPU executing time (441.92s) can be accepted when $popSize$ is 1280.

Moreover, we study the impact of parameter $subNum$ on convergence efficiency given $popSize=1280$ (see Fig 6, Table 4).

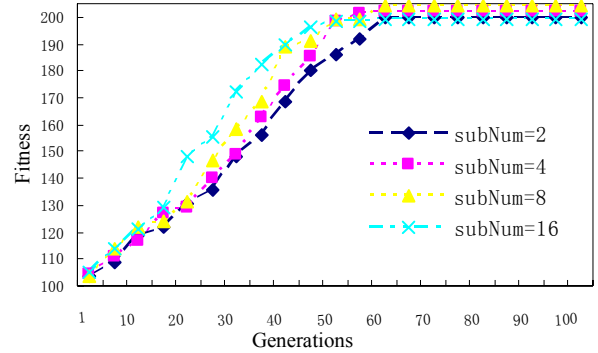


Figure 6 Fitness with $subNum$ changing

Fig 6 shows that convergence efficiency improves according to the increase of $subNum$. When $subNum$ reaches 8, convergence efficiency is the best and global optimal solution (204.08) can be gained. However, when $subNum$ is more than 16, convergence efficiency decreases obviously. The cause may be that size of subpopulation becomes too small if $subNum$ is too big, which make the sample lack of statistical characterization. Therefore, we argue that $subNum=8$ is reasonable.

V. CONCLUSION

This paper proposes a novel transmitter selection model in wireless network—PBCM based on [2] as well as a method to simulate this experiment. A new algorithm—PEDAs is proposed to solve the problem of wireless network planning. Experiments prove that our method outperforms commonly used algorithms in wireless network planning. Furthermore, convergence efficiency is observed along with different parameters.

REFERENCE

- [1] Patrice Calegaria, Frederic Guideca & Pierre Kuonena, et. Combinatorial optimization algorithms for radio network planning. Theoretical Computer Science 263 (2001) 235–245
- [2] Patrice Calegaria, Frederic Guideca & Pierre Kuonena, et. Parallel Island-Based Genetic Algorithm for Radio Network Design. JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING 47, 86–90 (1997)
- [3] Bahai, A.R.S.; Aghvami, H.; Network planning and optimization in the third generation wireless networks; 3G Mobile Communication Technologies, 2000. First International Conference on (IEE Conf. Publ. No. 471) 27–29 March 2000 Page(s):441 – 445
- [4] Qin Heren, Guan Lin, XieShengli; An Improved Genetic Algorithm for Transceiver Placement Problem in Radio Network Design, computer engineering and application, 2004(15) 72–73
- [5] Larranage & Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, 2001.
- [6] Endika Bengoetxea, Inexact Graph Matching Using Estimation of Distribution Algorithms, doctoral thesis, 2002:43–76.
- [7] Lienig J. A parallel genetic algorithm for performance-driven VLSI routing [J]. IEEE Trans on EC, 1997, 1 (1) :29–39.