A Cell Outage Compensation Mechanism in Self-Organizing RAN

Lingfeng Xia, Wenjing Li, Heng Zhang, Zhili Wang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications, Beijing, P.R.China
lfxia@bupt.edu.cn, wjli@bupt.edu.cn, st0rm@126.com, zlwang@bupt.edu.cn

***Abstract*—Cell Outage Management (COM) is a functionality aiming to automatically detect and mitigate outages that occur in radio networks due to unexpected failures. In this paper, we assume that future Radio Access Networks (RAN) can autonomously detect an outage based on measurements, then we propose a Genetic Algorithm (GA) based mechanism for Cell Outage Compensation (COC). COC is achieved by a GA-based COC mechanism. Simulation results showed that with GA, the network performance degradation is minimized.**

Keywords-self-healing; cell outage compensation; GA

I. INTRODUCTION

Self-organisation is among the hottest topics in telecommunication network research and development, eagerly awaited by network operators. The three commonly distinguished domains of self-organisation are self-configuration, self-optimisation, and self-healing. Self-healing methods aim to resolve the loss of coverage/capacity induced by such events as much as possible. A unified picture of these distinct components of self-organisation is given in Figure 1.

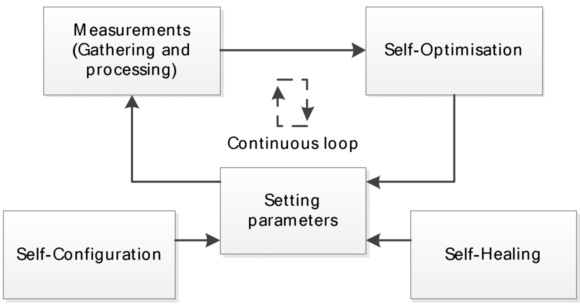

Figure 1. Self-organization in future mobile networks

Cell Outage Management (COM) comprises two functions: Cell Outage Detection (COD) and Cell Outage Compensation (COC), and is an integral part of the self-organizing network concept in E-UTRAN [2][3][4], with the objective to enhance the network robustness and resilience.

Figure 2 depicts the different elements and workflow of COM in future cellular networks. The depicted example is characterized by a site outage, whose pre-outage service area is indicated in red. A variety of measurements, e.g., alarms, counters or Key Performance Indicators (KPI), are gathered by the user terminals, the cells and/or the Operations Administration Maintenance (OAM) systems, and fed to the COD and COC [1]. Fed with these measurements, the COD function then automatically identifies the occurrence and scope of an outage, and triggers the COC function. The COC function translates these measurements into compensation measures in order to automatically mitigate the incurred coverage degradations, by an appropriate adaptation of one or more control parameters (e.g., antenna tilt, power settings) in surrounding cells.

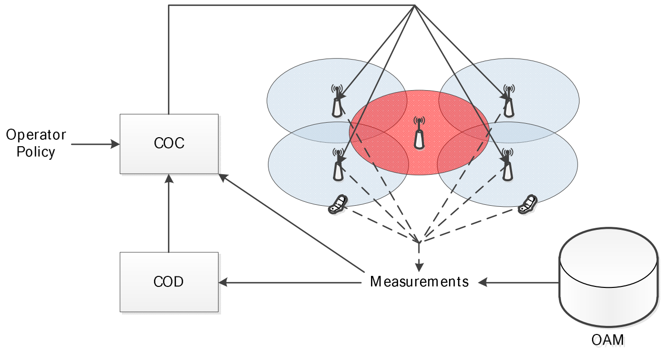

Figure 2. Overview of cell outage management

As depicted in Figure 3, after COC, some of the user equipments (UEs) served by the outage cell may be affected. This should be taken into account and an appropriate balance between the capacity/coverage offered to the outage area and the unavoidable performance degradation experienced in the surrounding cells, should be achieved. This balance is indicated by means of an operator policy that governs the actions taken by the cell outage compensation function.

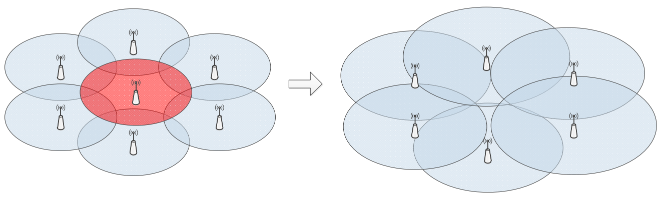

Figure 3. Description of COC

The remainder of this paper is organized as follows: section II describes related work; section III describes the GA-based COC Mechanism; simulation results are presented in Section IV.

This work was partly supported by the Funds for Creative Research Groups of China (60821001), 973 project of China (2007CB310703), NCET-10-0240 and National S&T Major Project (2011ZX03003-002-01).

COD has previously been studied and reported in literature. In [4], historic information is used in a Bayesian analysis to derive the probability of a cause (fault that initiates the problem) given the symptoms (manifestations of the causes). Knowledge of radio network experts is needed as well as databases from a real network in order to diagnose the problems in the network. In [5], a cell outage detection algorithm, which is based on the neighbor cell list reporting of mobile terminals is presented and evaluated.

In this work we propose a Genetic Algorithm (GA) based mechanism for cell outage management. Our objective is to minimize the network performance degradation when a cell is in outage through quick detection and compensation measures.

The COM has focused on the development of a set of cell outage compensation algorithms. GAs are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure, and apply recombination operators to these structures to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which GAs have been applied is quite broad. GAs are substantially different from more traditional search and optimization methods. The four most significant differences are:

- GAs search a population of points in parallel, not a single point.
- GAs do not require derivative information or other auxiliary knowledge; only the objective function and the corresponding fitness levels which influence the directions of the search.
- GAs use probabilistic transition rules, not deterministic ones.
- GAs work on an encoding of the parameter set rather than the parameter set itself (except when real-valued individuals are used).

It is important to note that the GA provides a number of potential solutions to a given problem and the choice of final solution is left to the user. In cases where a particular problem does not have one individual solution, for example a family of Pareto-optimal solutions, as is the case in multi-objective optimization and scheduling problems, the GA is potentially useful for identifying these alternative solutions simultaneously.

II. GA-BASED COC MECHANISM

A. Problem Formulation

Assume that m base stations (exclusive of the outage cell) are involved in a COC mechanism, and n users are severed by these base stations. The objective of COC is to minimize the sum of squares of the difference between the capacity utilization of a base station and the average capacity utilization of all the base stations. The constraints are: *1)* the distance between a base station and the served users should not exceed the maximum coverage distance, which is decided by uplink coverage of a UE (User Equipment); *2)* the capacity utilization of each base station should not exceed its threshold of capacity utilization; *3)* each user should be served by a base station.

The problem can be expressed as:

$$Minimize\ \ f(x) = \sum_{i=1}^{m}\left(\frac{\sum_{j=1}^{n} c_j}{\sum_{j=1}^{m} C_j} - \frac{\sum_{j=1}^{n} equal(x_j, i)\cdot c_i}{C_i}\right)^2$$

$$st. \begin{cases} \dfrac{\sum_{j=1}^{n} equal(x_j, i)\cdot c_i}{C_i} \le \eta_{ci}, \forall i \\ 1 \le x_i \le m, \forall i \end{cases} \qquad (1)$$

We call it problem (1), where C_j is the total capacity of bs_j, c_i is the capacity occupied by $user_i$, $\sum_{j=1}^{n} equal(x_j, i)\cdot c_{x_i}$ is the current capacity of bs_i, η_{cj} is the threshold of capacity utilization of bs_j (η_{cj} <1).

Solution to problem (1) is the expected COC mechanism, and can be expressed as a vector indicating the connection between users and base stations: $\boldsymbol{X}=[x_1,x_2,\ldots,x_n]$, where $1 \le x_i \le m$. We call $\boldsymbol{X}$ the connection vector. It can be proved that problem (1) is a NP hard problem. New solution is needed.

B. Genetic Algorithms

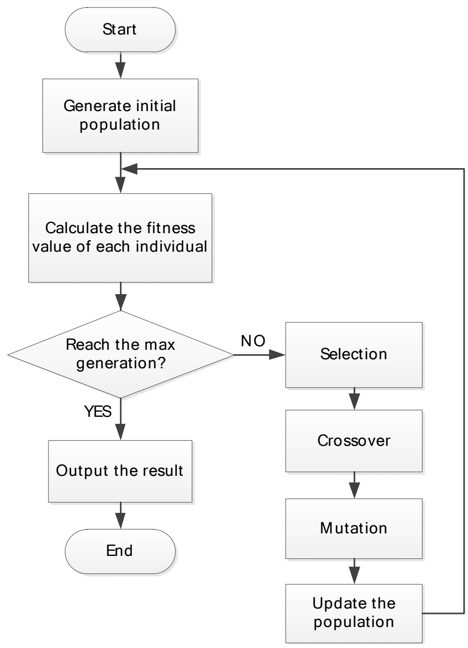

Figure 4. Procedure of GA

GAs operate on a population (a number of potential solutions), consisting of some encoding of the parameter set simultaneously. Here, each vector $\boldsymbol{X}=[x_1,x_2,\ldots,x_n]$ which indicates the connection between users and base stations is encoded as a string and these are concatenated to form a chromosome.

During selection we use stochastic universal sampling (SUS) which is a single-phase sampling algorithm with minimum spread and zero bias.

The basic operator for producing new chromosomes in the GA is that of crossover. Like its counterpart in nature, crossover produces new individuals that have some parts of both parent's genetic material. The section between the first allele position and the first crossover point is not exchanged between individuals, as depicted in Figure 5.

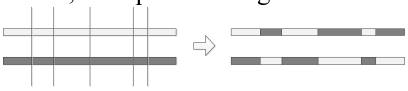

Figure 5. Crossover

Given a real-valued encoding of the chromosome structure, intermediate recombination is a method of producing new phenotypes around and between the values of the parents' phenotypes.

In natural evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. In GAs, mutation is randomly applied with low probability, and modifies elements in the chromosomes.

To maintain the size of the original population, the new individuals have to be reinserted into the old population. When selecting which members of the old population should be replaced the most apparent strategy is to replace the least fit members deterministically. Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria.

A common practice is to terminate the GA after a pre-specified number of generations and then test the quality of the best members of the population against the problem definition. If no acceptable solutions are found, the GA may be restarted or a fresh search initiated.

C. Objective and Fitness Functions

GAs operate on a number of potential solutions, called a population, consisting of some encoding of the parameter set simultaneously.

The objective function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the most fit individuals will have the lowest numerical value of the associated objective function. In the COC Mechanism, we defined the objective function as $f(x)$ in (1).

This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a GA. Another function, the fitness function, is normally used to transform the objective function value into a measure of relative fitness. In this mechanism, considering the constraints of the problem, we defined the fitness function $F(x)$ as follows:

$$F(x) = -f(x) \cdot P(x) \qquad (2)$$

$$P(x) = \frac{m}{\sum_{i=1}^{m} \varepsilon\left(\mathrm{C}_j \cdot \eta_{cj} - \sum_{j=1}^{n} equal(x_j, i) \cdot c_{x_j} \right)} \qquad (3)$$

$$\varepsilon(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$P(x)$ is a penalty function, which can transform a constrained problem to an unconstrained problem by penalizing the infeasible solutions. In (3), as the value of denominator gets smaller (in another way, some base stations' capacity utilization are greater than their threshold), $P(x)$ will get bigger, consequently, the corresponding fitness value $F(x)$ will get smaller. In that case, infeasible solutions will obtain a low fitness value, and the possibility they transmit to the next generation could be greatly reduced.

III. Simulation

In this paper, we choose MATLAB as our simulation environment. Parameters used in the simulation are listed in Figure 6.

Parameter	Value
NBS	10
NIND	50
MAXGEN	100
PRECI	1000
GGAP	0.9
CIND	$0 < CIND < 1$
Chrom	$[gene_1, gene_2, \ldots, gene_{NIND}]$
gen	0
P_c	0.3
P_m	0.1

Figure 6. Preferences in the simulation

In Figure 6, *Chrom* is a double dimensional array, which is consisted of *NIND* genes. Each gene equals a $\boldsymbol{X=[x_1,x_2,\ldots,x_n]}$ mentioned in the last section. *PRECI* is the number of users. *CIND* is a PRECI dimension vector, showing the capacity of each individual. If fewer individuals are produced by recombination than the size of the original population, the fractional difference between the new and old population sizes is termed a generation gap [7], we called it *GGAP* for short. Crossover probability, P_c, is set to 0.3. In GAs, mutation is randomly applied with low probability, P_m, here we set it to 0.1. The detailed algorithm is shown in Figure 7.

```
procedure GA
begin
    gen = 0;
    initialize Chrom(gen);
    calculate the objective value of Chrom(gen);
    while gen <=  MAXGEN do
    begin
        calculate the fitness value of Chrom(gen);
        select Chrom(gen) from Chrom(gen-1);
        mutate Chrom(gen);
        reinsert in Chrom(gen);
        calculate the objective value of Chrom(gen);
        gen = gen +1;
    end
end
```

Figure 7. Implementation of GA

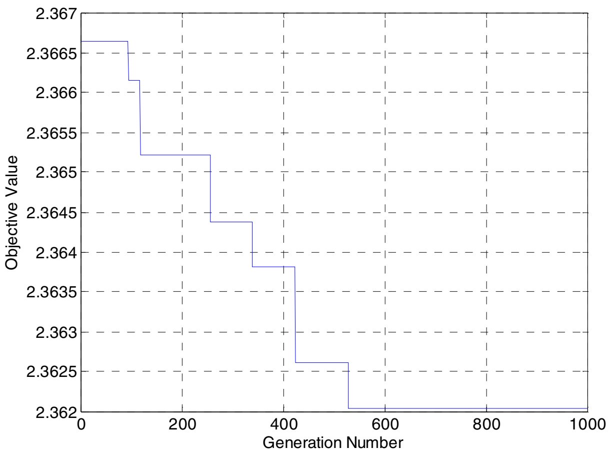

Figure 8. Objective value

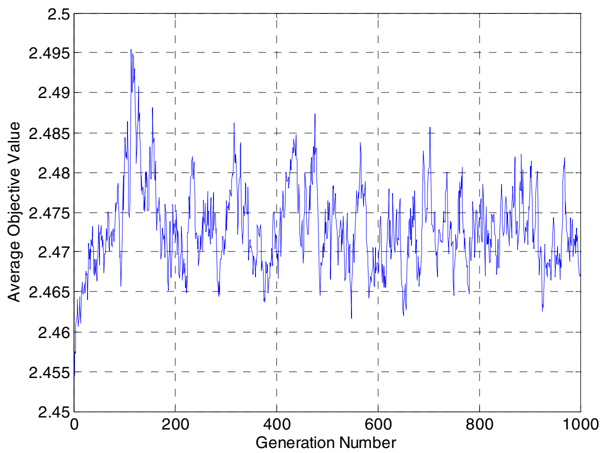

Figure 9. Average objective value

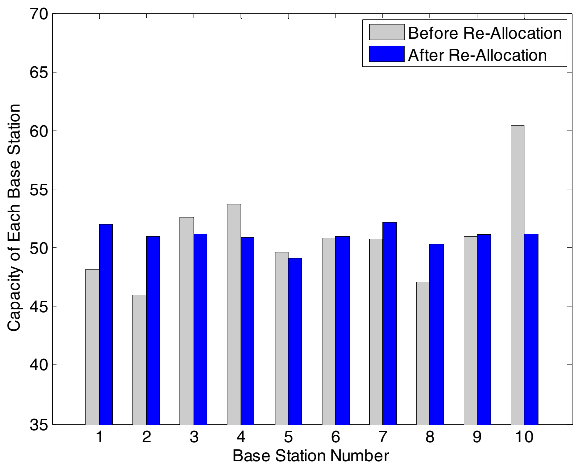

Figure 10. Variation of Base Stations' Capacity

Figures above show the performances of GA. In Figure 8, with the growing of the generation number, the minimum objective value (equals f(x) in problem (1))'s rate of descent gradually slow down. After nearly 460 generations, the minimum objective value comes to a stable level. As our main object, the sum of squares of the difference between the capacity utilization of a base station and the average capacity utilization of all the base stations can converge to the approximately optimal solutions. Meanwhile, as depicted in Figure 9, the average objective value converges faster than the minimum objective value. It takes less than 200 generations to reach the stable range. Figure 10 shows the variation of base stations' capacity, when the cell outage occurs, the rest normal base stations make corresponding adjustments to minimize the network performance degradation. After the re-allocation, capacity of the base stations achieve balanced load, compared with the condition before the re-allocation.

The results of the simulation demonstrate that the GA-based COC mechanism achieves a desired effect: objective value converges to the extent of the ideal, average objective value reaches a relatively stable value, and base stations mainly attain capacity load balance.

IV. CONCLUSION

In this work, COC is achieved in RAN by a GA-based COC mechanism. Simulation results demonstrated that with GA, the network performance degradation is minimized when a cell is in outage through quick detection and compensation measures. Efficiency of the GA is increased as there is no need to convert chromosomes to phenotypes before each function evaluation; less memory is required as efficient floating-point internal computer representations can be used directly; there is no loss in precision by discretization to binary or other values; and there is greater freedom to use different genetic operators. Our future work will focus on the management of COC in a specific network to promote the practicability and accuracy of our method. We will also create better simulations with richer impact factors and scenarios.

REFERENCES

[1] SOCRATES D5.9, "Final Report on Self-Organisation and its Implications in Wireless Access Networks", Version 1.0, 2010.

[2] 3GPP TR 36.902, "Self-configuring and self-optimizing network use cases and solutions", Version 9.2.0, 2010.

[3] 3GPP TS 32.541, "Self-Healing OAM; Concepts and Requirements", Version 1.2.0, 2010.

[4] M. Amirijoo, L. Jorguseski, T. Kürner, R. Litjens, M. Neuland, L. C. Schmelz, and U. Türke, "Cell Outage Management in LTE Networks", Proc. ISWCS '09, Siena, Italy, 2009.

[5] R. Barco, V. Wille, L. Diez, "System for Automated Diagnosis in Cellular Networks based on Performance Indicators", European Transactions on Telecommunications, Vol. 16, Issue 5, pp. 399 - 409, Sept. 2005.

[6] C. Mueller, M. Kaschub, C. Blankenhorn, S.Wanke, "A Cell Outage Detection Algorithm Using Neighbor Cell List Reports", International Workshop on Self-Organizing Systems, pp. 218 – 229, 2008..

[7] K. A. De Jong and J. Sarma, "Generation Gaps Revisited", In Foundations of Genetic Algorithms 2, L. D. Whitley (Ed.), Morgan Kaufmann Publishers, 1993.