

# Online Call Control in Cellular Networks Revisited

Yong Zhang\*    Francis Y.L. Chin<sup>†</sup>    Hing-Fung Ting<sup>‡</sup>    Joseph Wun-Tat Chan<sup>§</sup>  
Xin Han<sup>¶</sup>    Ka-Cheong Lam<sup>||</sup>

## Abstract

Wireless Communication Networks based on Frequency Division Multiplexing (FDM in short) play an important role in the field of communications, in which each request can be satisfied by assigning a frequency. To avoid interference, each assigned frequency must be different from the neighboring assigned frequencies. Since frequencies are scarce resources, the main problem in wireless networks is how to fully utilize the given bandwidth of frequencies. In this paper, we consider the online call control problem. Given a fixed bandwidth of frequencies and a sequence of communication requests arriving over time, each request must be either satisfied immediately after its arrival by assigning an available frequency, or rejected. The objective of the call control problem is to maximize the number of accepted requests. We study the asymptotic performance of this problem, i.e., the number of requests in the sequence and the bandwidth of frequencies are very large. In this paper, we give a  $7/3$ -competitive algorithm, say CACO, for the call control problem in cellular networks, improving the previous  $2.5$ -competitive result, and show that CACO is best possible among a class of HYBRID algorithms.

**Keywords:** Online algorithms, Call control problem, Cellular networks

## 1 Introduction

Frequency Division Multiplexing (FDM in short) is commonly used in wireless communications. To implement FDM, the wireless network is partitioned into small regions (cells) and each cell is equipped with a base station. When a call request arrives at a cell, the base station in this cell will assign a frequency to this request, and the call is established via this frequency. Since frequencies are scarce resources, to satisfy the requests from many users, a straightforward idea

---

\*Department of Computer Science, The University of Hong Kong, Hong Kong, yzhang@cs.hku.hk, Research supported by NSFC (11171086)

<sup>†</sup>Department of Computer Science, The University of Hong Kong, Hong Kong, chin@cs.hku.hk, Research supported by HK RGC grant HKU-7117/09E

<sup>‡</sup>Department of Computer Science, The University of Hong Kong, Hong Kong, hfting@cs.hku.hk, Research supported by HK RGC grant HKU-7171/08E

<sup>§</sup>College of International Education, Hong Kong Baptist University, Hong Kong, cswtchan@gmail.com

<sup>¶</sup>School of Software, Dalian University of Technology, China, hanxin.mail@gmail.com. Research supported by NSFC(11101065)

<sup>||</sup>Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, pandaman@163.com

is reusing the same frequency for different call requests. But if two calls which are close to each other are using the same frequency, interference will happen and that will violate the quality of communications. Thus, to avoid interference, the same frequency cannot be assigned to two different calls with close distance to each other. In general, the same frequency cannot be assigned to two calls in the same cell or neighboring cells.

There are two research directions on the fully utilization of the frequencies. One is the *frequency assignment problem*, and the other is *call control problem*. In the frequency assignment problem, each call request must be accepted, and the objective is to minimize the number of frequencies to satisfy all requests. In call control problem, the bandwidth of frequencies is fixed, thus, when the number of call requests in a cell or in some neighboring cells is larger than the total bandwidth, the request sequence cannot be totally accepted, i.e., some requests would be rejected. The objective of the call control problem is to accept as many requests as possible.

**Problem Statement:**

In this paper, we consider the online version of the call control problem. There are  $\omega$  frequencies available in the wireless network. A sequence  $\sigma$  of call requests arrives over time, where  $\sigma = \{r_1, r_2, \dots, r_t, \dots\}$ ,  $r_t$  denotes the  $t$ -th call request and also represents the cell where the  $t$ -th request arrives. When a request arrives at a cell, the system must either choose a frequency to satisfy this request without interference with other assigned frequencies in this cell and its neighboring cells, or reject this request. When handling a request, the system does not know any information about future call requests. We assume that when a frequency is assigned to a call, this call will never terminate and the frequency cannot be changed. This assumption does not reflect the world exactly, however, it is a basic case for the call control problem. The objective of this problem is to maximize the number of accepted requests.

We focus on the call control problem in cellular networks, in which each cell is a hexagonal region and has six neighbors, as shown in Figure 1. The cellular network is widely used in wireless communication networks.

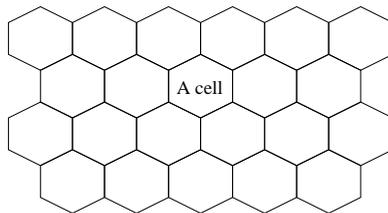


Figure 1: An example of a cellular network

**Performance Measure:**

To measure the performance of online algorithms, we use the competitive ratio [1] to compare the performance between the online algorithm and the optimal offline algorithm which knows the whole request sequence in advance. In the call control problem, the output is the set of accepted requests. For a request sequence  $\sigma$ , let  $A(\sigma)$  and  $O(\sigma)$  denote the number of accepted

request of an online algorithm  $A$  and the optimal offline algorithm  $O$ , respectively. We focus on the asymptotic performance for the call control problem, i.e., the number of requests and the number of frequencies are very large positive integers. The asymptotic competitive ratio for an online algorithm  $A$  is

$$R_A^\infty = \limsup_{n \rightarrow \infty} \max_{\sigma} \left\{ \frac{O(\sigma)}{A(\sigma)} \mid O(\sigma) = n \right\}.$$

### Related Work:

How to fully utilize the frequencies to satisfy the communication requests is a very fundamental problem in theoretical computer science and wireless communications. Both the frequency assignment problem and the call control problem are well studied during these years.

The offline version of the frequency assignment problem in cellular networks was proved to be NP-hard by McDiarmid and Reed [7], and two  $4/3$ -approximation algorithms were given in [7, 8]. In the online frequency assignment problem, when a call request arrives, the network must immediately assign a frequency to this call without any interference. There are mainly three strategies: Fixed Allocation [6], Greedy Assignment [2], and Hybrid Assignment [4]. If the duration of each call is infinity and the assigned frequency cannot be changed, the hybrid algorithm gave the best result for the online frequency assignment, i.e., a 2-competitive algorithm for the absolute performance and a 1.9126-competitive algorithm for the asymptotic performance.

For the call control problem, the offline version is NP-hard too [7]. To handle such problem, greedy strategy is always the first try, when a call request arrives, the network chooses the minimal available frequency to serve this request, if any frequency is interfere with some neighboring assigned frequency, the request will be rejected. Pantziou et al. [9] analyzed the performance of the greedy strategy, proved that the asymptotic competitive ratio of the greedy strategy is equal to the maximal degree of the network. Caragiannis et al. [2] gave a randomized algorithm for the call control problem in cellular networks, the asymptotic competitive ratio of their algorithm is 2.651. Later, the performance of the randomized algorithms was improved to  $16/7$  by the same authors [3], they also proved the lower bound of the asymptotic competitive ratio for the randomized algorithm is at least 2. Recently, a deterministic algorithm with asymptotic competitive ratio 2.5 was given in [10], and the lower bound of the asymptotic competitive ratio for the deterministic algorithm was proved to be 2.

### Our Contributions:

In this paper, we consider the deterministic algorithms for the online call control problem in cellular networks, and give a  $7/3$ -competitive algorithm, improving the previous 2.5-competitive result. Moreover, we define a class of algorithms, say HYBRID. Both the algorithm in [10] and ours are two special algorithms in HYBRID, and our algorithm is best possible among HYBRID.

## 2 Call Control in Cellular Networks

The idea of our algorithm for call control problem in cellular networks is similar to the algorithm in [10]. Cellular networks are 3 colorable, each cell can be associated with a color from  $\{R, G, B\}$

and any two neighboring cells are with different colors. Partition the frequencies into four sets,  $F_R$ ,  $F_G$ ,  $F_B$ , and  $F_S$ , where  $F_X$  ( $X \in \{R, G, B\}$ ) can be only used in cells with color  $X$  and  $F_S$  can be used in any cell. We define a class of algorithm, say **HYBRID**, for the online call control problem. In the HYBRID algorithms, when a request arrives at a cell with color  $X \in \{R, G, B\}$ , first choose the frequency from the set  $F_X$ , then from  $F_S$  if no interference appear. The performance are different w.r.t. the ratio between  $|F_R|$  ( $|F_G|$ ,  $|F_B|$ ) and  $|F_S|$ . Note that from symmetry, the size of  $F_R$ ,  $F_G$  and  $F_B$  should be same. Thus, the algorithm in [10] and ours are two special algorithms in HYBRID. By using a totally different analysis, the performance of our algorithm is better. Moreover, our algorithm is best possible among HYBRID.

Since we consider the asymptotic performance of the call control problem, we may regard the number  $\omega$  of frequencies in the system is a multiple of 7. Divide the frequencies into four disjoint sets as follows:

$$\begin{aligned} F_R &= \{1, \dots, 2\omega/7\}, \\ F_G &= \{2\omega/7 + 1, \dots, 4\omega/7\}, \\ F_B &= \{4\omega/7 + 1, \dots, 6\omega/7\}, \text{ and} \\ F_S &= \{6\omega/7 + 1, \dots, \omega\} \end{aligned}$$

Obviously, the ratio between the number of frequencies in  $F_R$ ,  $F_G$ ,  $F_B$ , and  $F_S$  is  $2 : 2 : 2 : 1$ .

Now we describe our algorithm CACO as follows:

---

**Algorithm 1 CACO:** When a request arrives at a cell  $C$  with color  $X \in \{R, G, B\}$

---

- 1: **if**  $F_X$  is not totally used up **then**
  - 2:     assign the minimal available frequency from  $F_X$  to satisfy this request.
  - 3: **else if**  $F_S$  is not totally used up in cell  $C$  and its neighboring cells **then**
  - 4:     assign the minimal available frequency from  $F_S$  to satisfy this request.
  - 5: **else**
  - 6:     reject this request.
  - 7: **end if**
- 

The idea to show the performance of our algorithm CACO is to prove that the ratio between the total number of accepted requests by CACO and the total number of satisfied requests by the optimal offline algorithm is at least  $3/7$ . To prove this, we analyze the number of satisfied requests in each cell and its neighboring cells, then compare the number with the optimum value.

Let  $R_i$  be the number of requests arrived in cell  $C_i$ . Let  $O_i$  be the number of requests accepted by the optimal offline algorithm in cell  $C_i$ .  $\sum_i O_i$  is the total number of accepted request by the optimal offline algorithm. Let  $A_i$  be the number of requests accepted by our online algorithm CACO in cell  $C_i$ .  $\sum_i A_i$  is the total number of accepted request by CACO. Let  $A_X(C_i)$  be the the number of requests accepted by CACO in cell  $C_i$  by assigning frequencies from frequency set  $F_X$ . It can be seen that  $A_i = A_R(C_i) + A_G(C_i) + A_B(C_i) + A_S(C_i)$ . If  $C_i$  is

colored with  $X \in \{R, G, B\}$ , then  $A_i = A_X(C_i) + A_S(C_i)$ .

**Fact 1.** *If  $R_i \geq 2\omega/7$  for cell  $C_i$ , we have  $O_i \leq R_i$ ,  $A_i \leq R_i$ , and  $A_i \geq 2\omega/7$ .*

According to the number of satisfied requests by the optimal offline algorithm, we classify the cells into two types: cell  $C_i$  is *safe* if  $O_i \leq 2\omega/3$ , and *dangerous* otherwise.

**Lemma 2.** *If cell  $C_i$  is safe, then  $A_i \geq 3O_i/7$ .*

*Proof.* Consider the following two cases.

- $R_i \leq 2\omega/7$

According to CACO, all requests in this cell must be satisfied when  $R_i \leq 2\omega/7$ , thus,  $A_i = R_i$ . Since  $R_i \geq O_i$ , we have  $A_i \geq 3O_i/7$ .

- $R_i > 2\omega/7$

In this case, CACO will accept at least  $2\omega/7$  requests by assigning frequencies from  $F_X$ , thus,  $A_i \geq 2\omega/7$ . Since  $C_i$  is safe,  $O_i \leq 2\omega/3$ , therefore, we have  $A_i \geq 3O_i/7$ .

Combining the above two cases, this lemma is true.  $\square$

**Lemma 3.** *No two dangerous cells are adjacent to each other, and a safe cell has at most 3 dangerous neighboring cells. All neighboring cells around a dangerous cell are safe.*

*Proof.* This fact can be proved by contradiction.

If a safe cell  $C$  has more than 3 dangerous neighboring cells, since  $C$  has 6 neighboring cells, there must exist two dangerous cells which are neighbors. From the definition of dangerous cell, the total number of accepted request in these two dangerous neighboring cells is strictly more than  $\omega$ .

Similarly, if a dangerous cell  $C'$  is a neighboring cell of another dangerous cell  $C$ , the total number of accepted requests in  $C$  and  $C'$  is strictly more than  $\omega$ .

In this way, we obtain a contradiction since in any adjacent cells, the total number of accepted requests is no more than  $\omega$ .  $\square$

According to the algorithm CACO, when a request cannot be satisfied in a cell  $C$  with color  $X$ , all frequencies in  $F_X$  must be used in  $C$ , and all frequencies in  $F_S$  must be used in  $C$  and its six neighbors. Thus, we have the following fact:

**Fact 4.** *If cell  $C$  cannot satisfy a request according to the algorithm CACO, then  $A_S(C) + \sum_{\text{all } k} A_S(C_k) \geq \omega/7$ , such that  $C_k$  is a neighboring cell of  $C$ .*

To compare the number of satisfied requests by CACO in each cell with the optimal offline solution, we define  $B_i$  as follows, where  $C_k$  represents a neighboring cell of  $C_i$ .

$$B_i = \begin{cases} 3O_i/7 & \text{if } C_i \text{ is safe} \\ A_i + \sum_{\text{all } k} (A_k - 3O_k/7)/3 & \text{if } C_i \text{ is dangerous.} \end{cases}$$

**Lemma 5.**  $\sum_i B_i \leq \sum_i A_i$ .

*Proof.* Suppose  $C_k$  is a safe cell. According to Lemma 2, we have  $A_k \geq 3O_k/7$ . From Lemma 3, we know that there are at most three dangerous neighbors around  $C_k$ , thus, after counting  $B_k = 3O_k/7$  frequencies in  $C_k$ , the remaining  $A_k - 3O_k/7$  frequencies can compensate the frequencies in its dangerous neighbors, and each dangerous cell receives  $(A_k - 3O_k/7)/3$  frequencies. From the definition of  $B_i$ , we can see that  $\sum_i B_i \leq \sum_i A_i$ .  $\square$

**Theorem 6.** *The asymptotic competitive ratio of algorithm CACO is at most 7/3.*

*Proof.* From the definition of  $O_i$  and  $B_i$ , we can say that  $O_i/B_i \leq 7/3$  for any cell  $C_i$  leads to the correctness of this theorem. That is because

$$\frac{\sum_i O_i}{\sum_i A_i} \leq \frac{\sum_i O_i}{\sum_i B_i} \leq \max_i \frac{O_i}{B_i}.$$

If cell  $C_i$  is safe, i.e.,  $O_i \leq 2\omega/3$ , we have  $O_i/B_i = 7/3$ .

If cell  $C_i$  is dangerous, i.e.,  $O_i > 2\omega/3$ , since  $R_i \geq O_i > 2\omega/3 > 3\omega/7$  and  $A_i \leq 2\omega/7 + \omega/7 = 3\omega/7$ , the number of requests  $R_i$  in this cell is larger than  $A_i$ . Thus, some requests are rejected, and this cell cannot accept any further requests.

- If the number of accepted requests in any neighbor of  $C_i$  is no more than  $2\omega/7$ , that means all frequencies in  $F_S$  are assigned to requests in cell  $C_i$ . Thus,  $A_i = 3\omega/7$ . In this case, we have

$$O_i/B_i = O_i / (A_i + (\sum_k (A_k - 3O_k/7))/3) \leq O_i/A_i \leq \omega/A_i = 7/3.$$

Note that cell  $C_k$  is safe since cell  $C_i$  is dangerous.

- Otherwise, suppose there are  $m$  neighbors of  $C_i$  in which the number of accepted requests are more than  $2\omega/7$ . Let  $\hat{O}_i$  denote the average number of the optimum value of accepted requests in these  $m$  neighboring cells around  $C_i$ .

$$\begin{aligned}
B_i &= A_i + \left( \sum_k (A_k - 3O_k/7) \right) / 3 \\
&= 2\omega/7 + A_S(C_i) + \left( \sum_k (A_k - 3O_k/7) \right) / 3 \\
&\geq 2\omega/7 + A_S(C_i) + \left( m \times 2\omega/7 + \sum_{\substack{\text{for the neighbors} \\ \text{with } A_k > 2\omega/7}} A_S(C_k) - m \times 3\hat{O}_i/7 \right) / 3 \\
&\geq 2\omega/7 + \left( m \times 2\omega/7 + \sum_{\substack{\text{for the neighbors} \\ \text{with } A_k > 2\omega/7}} A_S(C_k) + A_S(C_i) - m \times 3\hat{O}_i/7 \right) / 3 \\
&\geq 2\omega/7 + (m \times 2\omega/7 + \omega/7 - m \times 3\hat{O}_i/7) / 3 \\
&\geq 2\omega/7 + (2\omega/7 + \omega/7 - 3\hat{O}_i/7) / 3 \\
&\quad \text{(that is because for any neighbor with } A_k > 2\omega/7, O_k \leq (\omega - O_i) \leq \omega/3 \\
&\quad \text{since cell } C_i \text{ is dangerous, thus, } \hat{O}_i \leq \omega/3 \text{ and } 2\omega/7 - 3\hat{O}_i/7 \geq 0.) \\
&\geq 2\omega/7 + (3\omega/7 - 3(\omega - O_i)/7) / 3 \\
&\quad \text{(since } O_k \leq \omega - O_i, \text{ we have } \hat{O}_i \leq \omega - O_i) \\
&= 2\omega/7 + O_i/7
\end{aligned}$$

Thus,  $O_i/B_i \leq O_i/(2\omega/7 + O_i/7) \leq 7/3$ .

From the above analysis, we can say that the asymptotic competitive ratio of the algorithm CACO is at most  $7/3$ .  $\square$

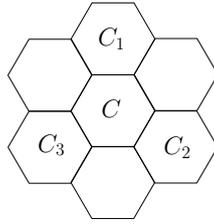


Figure 2: Algorithm CACO is best possible among HYBRID.

**Theorem 7.** *CACO is best possible among HYBRID.*

*Proof.* Suppose in a HYBRID algorithm,  $|F_R| : |F_G| : |F_B| : |F_S| = x : x : x : y$ . Consider the configuration shown in Figure 2.

In the first step,  $\omega$  requests arrive at the center cell  $C$  with color  $X$ , the algorithm will use up all frequencies in  $F_X$  and  $F_S$ . In this case, the online algorithm accepts  $(x + y)\omega/(3x + y)$  requests while the optimal offline algorithm may accept all  $\omega$  requests. Thus, the ratio between the optimal offline algorithm and the online algorithm is at least  $(3x + y)/(x + y)$  since the optimal algorithm will accept all these requests.

In the second step,  $\omega$  requests arrive at each  $C_1$ ,  $C_2$  and  $C_3$  with the same color. The online algorithm can only accept  $x\omega/(3x+y)$  requests in each  $C_i$  ( $1 \leq i \leq 3$ ) since the frequencies in  $F_S$  are all used in  $C$ . The online algorithm accepts  $(x+y)\omega/(3x+y) + 3x\omega/(3x+y) = (4x+y)\omega/(3x+y)$  requests in these two steps while the optimal offline algorithm will accept all  $3\omega$  requests in  $C_i$  ( $1 \leq i \leq 3$ ) and reject all requests in  $C$ . Thus, the ratio between the optimal offline algorithm and the online algorithm is at least  $3(3x+y)/(4x+y)$ .

Balancing these two ratios, we have  $x : y = 2 : 1$ . From the description of the above two steps, the lower bound of competitive ratio of HYBRID is  $7/3$ .  $\square$

## References

- [1] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis* Cambridge University Press, 1998.
- [2] Ioannis Caragiannis, Christos Kaklamanis, and Evi Papaioannou. Efficient on-line frequency allocation and call control in cellular networks. *Theory Comput. Syst.*, 35(5):521-543, 2002. A preliminary version of the paper is in SPAA 2000.
- [3] Ioannis Caragiannis, Christos Kaklamanis, and Evi Papaioannou. Competitive Algorithms and Lower Bounds for On-Line Randomized Call Control in Cellular Networks. *Networks* 52(4): 235-251, 2008. Preliminary versions are in WAOA03 and EUROPAR05.
- [4] Wun-Tat Chan, Francis Y.L. Chin, Deshi Ye and Yong Zhang. Online Frequency Allocation in Cellular Networks. In *Proc. of the 19th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2007)*, pp. 241-249.
- [5] Frédéric Havet. Channel assignment and multicoloring of the induced subgraphs of the triangular lattice. *Discrete Math.* 233, 219-231 (2001).
- [6] V. H. MacDonald. Advanced mobile phone service: The cellular concept. *Bell Systems Technical Journal*, 58(1):15-41, 1979.
- [7] Colin McDiarmid and Bruce Reed. Channel assignment and weighted coloring. *Networks*, 36(2):114-117, 2000.
- [8] Lata Narayanan, and Sunil Shende. Static frequency assignment in cellular networks. *Algorithmica*, 29(3):396-409, 2001.
- [9] Grammati E. Pantziou, George P. Pentaris, and Paul G. Spirakis. Competitive Call Control in Mobile Networks. *Theory of Computing Systems*, 35(6): 625-639, 2002.
- [10] Deshi Ye, Xin Han, and Guochuan Zhang. Deterministic On-line Call Control in Cellular Networks. *Theor. Comput. Sci.* 411(31-33): 2871-2877 (2010)
- [11] Yong Zhang, Francis Y.L. Chin, and Hong Zhu. A 1-Local Asymptotic  $13/9$ -Competitive Algorithm for Multicoloring Hexagonal Graphs. *Algorithmica* (2009) 54:557-567.