經營科學
第23卷 第3號
2006年 11月

차세대 이동통신에서의 지연을 고려한 순차적 페이징

†이채영* · 구상훈*

Sequential Paging under Delay Bound for Next Generation Mobile Systems

Chae Yong Lee* · Sang Hoon Ku*

■ Abstract ■

To reduce the signaling cost of paging in mobile communication, sequential paging schemes are proposed by partitioning a location area into several paging areas such that each area is paged sequentially. Necessary conditions for the optimal partition of cells with delay bound are examined by considering the mobiles location probability at each cell. The Optimal Cell Partitioning (OCP) is proposed based on the necessary conditions and the fathoming rule which trims off the unnecessary solution space and expedite the search process. Two Heuristics, BSG and BNC are also presented to further increase the computational efficiency in real-world paging scheme for the next generation mobile systems.

The effectiveness of the proposed paging schemes is illustrated with computational results. The Heuristic BSG that performs the search in the most promising solution group outperforms the best existing procedure with the 6-69% gain in paging cost in problems with 100 cells.

Keyword : Mobile Communication, Sequential Paging, Delay Bound

I. Introduction

One of the important issues in wireless network is the design and analysis of strategies for tracking the mobile terminals (MTs). Mobility tracking is concerned with finding an MT within

논문접수일 : 2006년 07월 12일 논문게재확정일 : 2006년 10월 18일
* 한국과학기술원 산업공학과
† 교신저자

the area serviced by the wireless network. Two basic operations for tracking an MT are : *location update* and *paging*. When an MT enters a new Location Area (LA), it performs a location update via the access channel to a base station in the new LA. Consequently, the system is always aware of the current location of an MT. Paging is the process in which a system searches for an MT by sending polling messages to the cells within the last reported LA of the MT.

The current GSM and IS-41 protocols perform a broadcast paging procedure [2, 3, 12] in which the mobile switching center (MSC) broadcasts the paging request to all cells in the MT's last registered LA. All base stations within the same LA broadcast the identifier (ID) of their LA periodically. Each MT compares its registered LA ID with the current broadcast LA ID. Location update is triggered if the two IDs are different. Upon a call arrival for a particular MT, all cells within its current LA are polled simultaneously, ensuring paging success within a single step.

Since the paging signal is broadcast at every base station in the LA, the signaling cost can be measured in terms of the number of cells to be searched before the called MT is found. Thus, considering the resource utilization [4, 7], the current broadcast paging scheme is inefficient since all cells in the LA are searched. Moreover, as the size of an LA increases with microcells, a significant amount of radio resource is consumed in paging for each call arrival. This cannot be scalable to next generation mobile systems with the growing number of mobile users and a variety of service characteristics which requires increased signaling and processing load in the wireless network.

In order to improve the efficiency of bandwidth utilization, many sequential paging schemes under the delay constraints are suggested [1-2, 6-12]. With the constraint of delay bound the minimization of paging costs requires the partitioning of an LA into several paging areas based on the location probability of each cell. Then each paging area is searched sequentially within the delay bound. Thus the essence of the minimization is how to partition the cells into paging areas. Goodman et al. [6] propose a grouping algorithm based on the dynamic programming. They partition cells into groups by iteratively increasing the number of paging areas up to the delay bound. Since the computation is based on the dynamic programming, the well-known curse of dimensionality problem exists as the number of cells and the delay bound increase.

Wang et al. [10] starts the partitioning procedure first by sorting the cells into nonincreasing location probabilities. Those cells are then evenly distributed into D paging areas with preference to paging area D, D-1, ⋯ for remaining cells. The procedure continues by moving cells in a paging area back and forth by the boundary conditions. The performance of the procedure is very efficient compared to other search procedures [10] in the literature. However, the procedure cannot guarantee the optimal partition of the cells that minimizes the paging cost within the delay bound. This is because the procedure starts by distributing cells evenly into each paging area.

In this paper, we investigate the necessary conditions for the optimal partition of cells into paging areas with delay bound. An optimal partition is provided based on the necessary conditions and a fathoming rule that eliminates unnecessary solutions in the search process. Two heuristics are also examined to reduce the search process of the optimal partition. Computational result shows that the partitioning algorithm and the two heuristics provide better solution quality compared to the procedure by [10].

2. Minimization of Paging Cost with Delay Bound

We assume that coverage area of the cellular network is divided into LAs and that the probability an MT is residing at each cell is given. Also, assume the paging cost C is measured in terms of the number of cells paged before the called MT is found. The delay bound D is measured in terms of the number of polling cycles [10]. For instance, if D=1, the system should find the called MT in one polling cycle, requiring all cells in the LA to be polled simultaneously. In this case the paging cost C is equal to the total number of cells N in the LA. We consider a partition x of cells such that $1 \le D \le N$, which requires grouping cells in an LA into D paging areas. The location probability p is given by $p=[p_1, p_2, p_3, \cdots, p_N]$, where p_j is the probability that the called MT is found in cell j and satisfies $p_1 \ge p_2 \ge \cdots \ge p_N$. Given a partition x, Let n_i be the number of cells contained in the paging area i, and q_i be the probability that the called MT is found in the paging area i. Clearly, the location probability q_i of the paging area i is given by

$q_i = \sum_{j \in PA(i)} p_j$, where $PA(i)$ is paging area i.

The *expected paging cost* of a partition x under delay bound D, $E[C(x)]$, is computed as follows:

$$E[C(x)] = \sum_{i=1}^{D} q_i \cdot k_i, \text{ where } k_i = \sum_{k=1}^{i} n_k$$

Also, the expected delay of the solution x, $E[D(x)]$, is

$$E[D(x)] = \sum_{i=1}^{D} i \cdot q_i$$

2.1 Necessary Conditions for Optimal Partition of Cells

To minimize the paging cost under delay bound D, the following necessary conditions and a corollary are proposed.

Lemma 1.
The expected paging cost $E[C(x)]$ is minimized, when the number of paging areas is equal to the delay bound D.

Proof. Consider a partition x with d paging areas, $1 \le d \le D-1$. To prove the lemma, it is enough to show that there exists a partition x' which has lower paging cost than x with d+1 paging areas. Let x' be a new partition generated from x by dividing an arbitrary paging area l into two paging areas l and l+1 such that

$$n_i = n_i^{'},\ q_i = q_i^{'} \qquad i = 1, \cdots, l-1$$

$$n_i = n_l^{'} + n_{l+1}^{'},\ q_l = q_l^{'} + q_{l+1}^{'}$$

$$n_i = n_{i+1}^{'},\ q_i = q_{i+1}^{'} \qquad i = l+1, l+2, \cdots, d$$

where n'_i and q'_i are respectively the number of cells and the probability that the called MT is found in the paging area i in partition x'. Then the average paging cost of x and x' are computed as follows:

$$E[C(x)] = [\sum_{i=1}^{l-1} q_i \cdot k_i + q_l(k_{l-1} + n_l) + \sum_{i=l+1}^{d-1} q_i \cdot k_i]$$

$$E[C(x')] = [\sum_{i=1}^{l-1} q_i \cdot k_i + q'_1(k_{l-1} + n'_l) +$$

$$q'_{l+1}(k_{l-1} + n'_{l+1}) + \sum_{i=l+2}^{d} q_i \cdot K_i]$$

Since $\sum_{i=l+1}^{d-1} q_i \cdot k_i = \sum_{i=l+2}^{d} q_i \cdot k_i$ in the above two equations, we have

$$\begin{aligned} & E[C(x)] - E[C(x')] \\ & = q_l(k_{l-1} + n_l) - [q_l(k_{l-1} + n'_l) + q'_{l+1} \cdot n'_{l+1}] \\ & = q_l(n_l - n'_l) - q'_{l+1} \cdot n'_{l+1} \\ & = q'_l \cdot n'_{l+1} > 0 \end{aligned}$$

It shows that the cost of x' with d+1 paging areas is less than the cost x with d paging areas.

Lemma 2. [Rose and Yates, 6]
The expected paging cost $E[C(x)]$ is minimized, when the cells are paged in nonincreasing order of their location probabilities.

Lemma 3.
Except the case where all location probabilities of the cells are equal, the expected paging cost $E[C(x)]$ is minimized when the paging area is paged in the order of nondecreasing number of cells in each area.

Proof. Consider an optimal partition x where $n_l \geq n_{l+1} + 1$ for an arbitrary paging area l, l=1, 2, ⋯, D-1. Let x' be a new partition obtained by moving the cell with the smallest probability in paging area l of partition x to paging area l+1. Let the smallest probability be p^s, then we have

$$E[C(x)] - E[(x')]$$

$$= [\sum_{i=1}^{l-1} q_i \cdot k_i + q_l(k_{l-1} + n_l) + q_{l+1}(k_{l-1} + n_l + n_{l+1})$$

$$+ \sum_{i=l+2}^{D} q_i \cdot k_i] - [\sum_{i=1}^{l-1} q_i \cdot k_i + (q_l - p^s)(k_{l-1} + n_l - 1)$$

$$+ (q_{l+1} + p^s)(k_{l-1} + n_l + n_{l+1}) + \sum_{i=l+2}^{D} q_i \cdot k_i]$$

$$= q_l - p^s(n_{l+1} + 1)$$

$$> p^s \cdot n_l - p^s(n_{l+1} + 1) \geq 0$$

This is a contradiction to that x is an optimal partition. Therefore, $n_i \leq n_{i+1}$ is satisfied for all i =1, ⋯, D-1 in the optimal partition. In fact, $E[C(x)] = E'[C(x')]$, if all probabilities in paging area l are equal to p^s and $n_l = n_{l+1} + 1$.

Corollary 1. (Another expression of Lemma 3) To satisfy the Lemma 3, the number of cells n_i in the paging area i is determined as follows:

$$1 \leq n_1 \leq \left[\frac{N}{D}\right] \text{ and } n_{i-1} \leq n_i \leq \left[\frac{n - \sum_{k=1}^{i-1} n_k}{D - i + 1}\right] \text{ for}$$

$i = 2, 3, ..., D$..

2.2 Partition of Cells into Paging Areas

Throughout this paper we assume cells are sequenced in nonincreasing order of location probability. Then by *Lemma* 1, 2, and 3, to find a solution x is to partition the cells into D paging areas such that the number of cells in the D paging areas is nondecreasing in the order of paging sequence. Thus the problem becomes to find the number of cells in each paging area without changing the cell sequence of the decreasing location probability. More specifically, let $x = (n_1, n_1, \cdots, n_{D-1}, n_D)$ where n_i is the number of cells in paging area i, then from Lemma 3 it is clear that $n_1 \leq n_2 \leq ... \leq n_{D-1} \leq n_D$, when not all the location probabilities are the same in a paging area. Now, our problem is to find a partition x that minimizes the expected paging cost $E[C(x)]$ while satisfying the Lemma 3. To find the optimal partition, we classify solutions depending on the solution group $k-1, \ldots, K$ and the number of cells m in the paging area D-1. The solution group k is defined as a set of all solutions that have the same number of cells in each paging area from 1 to D-2. Let x_m^k be a solution in group k with m cells in the paging area D-1. Then the following fathoming condition is satisfied.

Fathoming Rule: For a solution x_m^k in the solution group k, if $E[C(x_{m+1}^k)] \geq E[C(x_m^k)]$ then the solution group k is fathomed, which means that x_m^k gives the minimum cost in the solution group k. Hence solutions $x_{m+2}^k, x_{m+3}^k \cdots$ need not be evaluated any further.

Proof. Suppose that the paging cost of x_{m+1}^k is greater than or equal to that of x_m^k, and that there exists a solution x_{m+t}^k for $t \geq 2$ which has lower paging cost than x_m^k. Let $p^1, p^2, \ldots, p^t$ be respectively the largest, the second largest and the t-th largest probability in paging area D of x_m^k. Since $E[C(x_m^k)] - E[C(x_{m+1}^k)] \leq 0$, we have

$$E[C(x_m^k)] - E[C(x_{m+1}^k)]$$

$$= [\sum_{i=1}^{D-2} q_i \cdot k_i + q_{D-1} \cdot (k_{D-2}+m) + q_D \cdot N]$$

$$-[\sum_{i=1}^{D-2} q_i \cdot k_i + (q_{D-1}+P^1)(k_{D-2}+m+1) + (q_D - p^1) \cdot N]$$

$$= -q_{D-1} + p^1(n_{D-1} + n_D - m - 1)$$

$$= -q_{D-1} + p^1(n_D - 1) \leq 0$$

Thus, we have

$$p^1 \leq \frac{q_{D-1}}{n_D - 1}$$

Now, for any $t \geq 2$

$$E[C(x_m^k)] - E[C(x_{m+1}^k)]$$

$$= [\sum_{i=1}^{D-2} q_i \cdot k_i + q_{D-1} \cdot (k_{D-2}+m) + q_D \cdot N]$$

$$-[\sum_{i=1}^{D-2} q_i \cdot k_i + (q_{D-1} + p^1 + p^2 + \cdots + p^t)$$

$$(k_{D-2} + m + t) + (q_D - p^1 - p^2 - \cdots - p^t) \cdot N]$$

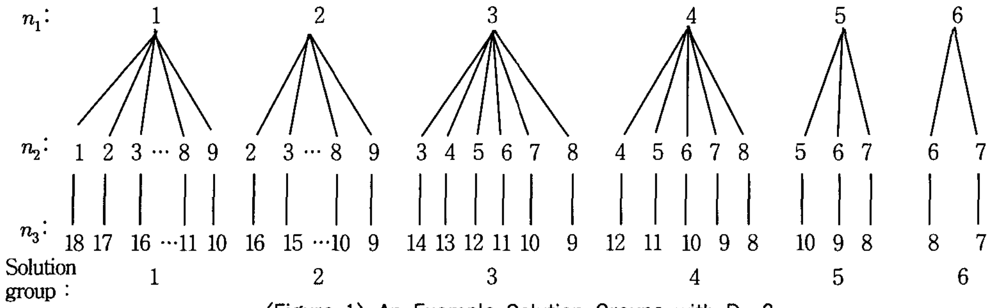
$$= -tq_{D-1} + (p^1 + \cdots + p^t)(n_D - t)$$

$$\leq -t\, q_{D-1} + t\, p^1 (n_D - t)$$

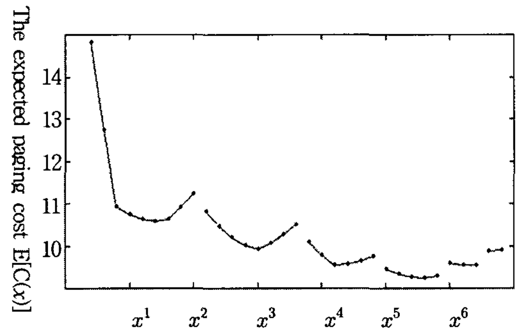
$$\leq t\, q_{D-1} \left[\frac{n_D - 1}{n_D - t} - 1 \right] < 0$$

Therefore, we have $E[C(x_{m+1}^k) > E[C(x_m^k)]$ which is a contradiction to the assumption that there exists a solution x_{m+1}^k which has lower paging cost than x_m^k. Thus, the solution group k is fathomed, if $E[C(x_{m+1}^k) \geq E[C(x_m^k)]$.

As an example, consider a problem with N =20, D=3 and p = [0.14, 0.14, 0.13, 0.13, 0.04, 0.04, 0.04, 0.04, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025]. <Figure 1> shows 33 feasible solutions that satisfy the delay constraint. The 33 solutions are classified into six solution groups depending on the number of cells in the paging area 1. In the figure solution group 1 has nine feasible solutions x_1^1 = (1, 1, 18), x_2^1 = (1, 2, 17), ⋯, and x_9^1 = (1, 9, 10) depending on the number of cells in the paging area 2. <Figure 2> shows the paging cost of each solution in the same order shown in <Figure 1>. According to the fathoming rule,



<Figure 1> An Example Solution Groups with D=3



<Figure 2> The Paging Cost of Solutions in the Six Groups

the solution x_6^1 in <Figure 2> gives the minimum paging cost in the solution group 1. In other words, since $E[C(x_6^1)] < E[C(x_7^1)]$, solution group 1 is fathomed and x_8^1 and x_9^1 need not be evaluated any further. The fathoming rule is also applied to other solution groups and reduces the computational effort by cutting out unnecessary partitions.

3. Cell Partitioning Algorithms for Sequential Paging

To find a solution that minimizes the expected paging cost under the delay bound, threealgorithms are developed by employing necessary conditions and the fathoming rule presented in Section II.

3.1 Optimal Cell Partitioning

This algorithm is designed to give the optimal partition of cells that minimizes the paging cost. All feasible solutions x_m^k that satisfy the three necessary conditions are generated and evaluated at each solution group $k=1, \cdots, K$. The fathoming rule is applied to reduce the solution space. Before introducing the algorithm, notice that the minimum and the maximum number of cells in paging area $D-1$ are different for each solution group k. Let n_i^k be the number of cells in paging area i of solution group k. Then from Lemma 3, it is clear that

$$n_{D-2}^k \le n_{D-1}^k \le \left[\frac{N - \sum_{i=1}^{D-2} n_i^k}{2} \right]$$

To simplify the notation, we denote $n_{\min}^k = n_{D-2}^k$ and $n_{\max}^k = \left[\frac{N - \sum_{i=1}^{D-2} n_i^k}{2} \right]$.

Algorithm OCP

Step 1 : Sort the cells in nonincreasing order of location probability by Lemma 2.

Step 2 : Generate solutions x_m^k for $k=1, 2, \cdots, K$ and $n_{\min}^k \le m \le n_{\max}^k$ by Lemma 1 and 3. Let $k=0$ and $z_k \leftarrow \infty$.

Step 3 : Let $k \leftarrow k+1$, if $k > K$, then go to Step 4. Let $m \leftarrow n_{\min}^k$ and $E[C(x_{m-1}^k)] \leftarrow \infty$. Compute $E[C(x_m^k)]$.

3.1 If $E[C(x_m^k)] < E[C(x_{m-1}^k)]$, then, $z_k = E[C(x_m^k)]$, $x_{opt}^k = x_m^k$ and $m \leftarrow m+1$.
If $m > n_{\max}^k$, let $z_{opt} = z_k$ and $x_{opt} = x_{opt}^k$, if $z_k < z_{k-1}$. Go to Step 3. Otherwise, repeat 3.1.

3.2 If $E[C(x_m^k)] \geq E[C(x_{m-1}^k)]$, then the solution group k is fathomed. If $z_k < z_{k-1}$, then $z_{opt} = z_k$, $x_{opt} = x_{opt}^k$ and go to Step 3.

Step 4 : Stop with the optimal partition x_{opt} and the optimal paging cost z_{opt}.

3.2 Heuristic with the Best Solution Group

To reduce the computational effort required in the Algorithm OCP we propose a heuristic based on the best solution group. To select the best group a solution in each group is evaluated and compared. In each group the solution with the minimum number of cells in paging area $D-1$ is selected and compared. The group k' that gives the minimum paging cost is further investigated. The best solution is obtained by applying the fathoming rule in the solution group k'.

Heuristic BSG

Step 1 : Sort the cells in nonincreasing order of location probability by Lemma 2.

Step 2 : Generate solutions x_m^k for $k=1, 2, \cdots, K$ and $n_{\min}^k \leq m \leq n_{\max}^k$ by Lemma 1 and 3.

Step 3 : Evaluate solutions x_m^k for $k=1, 2, \cdots, K$ and $m = m_{\min}^k$.
Choose the partition with the smallest paging cost. Let x_{best} and z_{best} be the corresponding partition and the paging cost respectively. Also let k' be the corresponding solution group. Let $m \leftarrow n_{\min}^{k'} + 1$.

Step 4 : If $E[C(x_m^{k'})] < E[C(x_{m-1}^k)]$, then $z_{best} = E[C(x_m^{k'})]$ and $x_{best} = x_m^{k'}$. Let $m \leftarrow m+1$ and repeat this step.

Otherwise, the solution group k' is fathomed. Stop with the best partition x_{best} and the paging cost z_{best}.

The Heuristic BSC, if applied to the example problem of Section II-B, first evaluates $x_1^1, x_1^2, \cdots$, and x_1^6 in <Figure 2>. Since x_1^4 gives the minimum cost, solution group $k'=4$ is selected and further investigated to have the solution x_4^4.

3.3 Heuristic with the Best Number of Cells in Each Paging Area

The heuristic is designed based on the Corollary 1. The number of cells n_i for each paging area i is determined such that $n_1 \leq n_2 \leq \cdots n_D$. The uniform paging [Wang et al., 11] concept is employed for the number of cells in the remaining paging area. At each step n_i^* with the minimum paging cost is selected among n_i that satisfy Corollary 1.

Heuristic BNC

Step 1 : Sort the cells in nonincreasing order of location probability by Lemma 2.

Step 2 : For paging area i, determine the number of cells n_i^* as follows (assume $n_i^*, n_2^*, \cdots, n_{i-1}^*$ is determined):

2.1 For each n_i that satisfies Corollary 1

Let $n_0 = \left\lfloor \frac{N - \sum_{k=1}^{i} n_k}{D-i} \right\rfloor$

where $N - \sum_{k=1}^{i} n_k = n_0(D-i) + l$.

Note that l is the number of cells remaining after n_0 cells are evenly assigned to the $D-i$ paging areas. Let $n_{i+1} = n_{i+2} = \cdots = n_{D-1} = n_0$ and $n_{D-l+1} = \cdots = n_D = n_0 + 1$. Compute the paging cost of the partition.

2.2 Select the partition that minimizes the paging cost and let n_i^* be the n_i with the minimum paging cost.

4. Computational Results and Discussion

In this section, computational experiment is performed to examine the effectiveness of the proposed paging algorithms. The algorithms presented in Section III are implemented in Matlab version 5.2.0, and run on a 700MHz Intel Pentium III based personal computer with 256Mbyte of memory under Windows 98.

Three problem sets are generated each with 30, 50, and 100 cells. In each set ten different problems are experimented. The cell location probabilities are obtained by assuming the-following geometric distribution:

$$p_j = r^j / \sum_{j=1}^{N} r^j \text{ for } 0 < r \le 1$$

The above cell location probabilities may well suggest some real world situations [5], where user location probabilities are concentrated in a relatively small portion of the area when r has a low value. In each problem set nine problems are generated each with $r = 0.1, 0.2, \cdots$, and 0.9. In the last problem of each set the cell location probabilities are assumed to follow uniform distribution.

<Table 1~3> show the computational result of the Algorithm OCP, Heuristic BSG, Heuristic BNC and the algorithm by [10]. Note in our experiment that when the delay bound $D=2$, all four procedures give the same solution in every problem. Thus the computational results with $D=3$, 4 and 5 are presented.

From <Table 1> it is clear that Heuristic BSG and BNC lead to the same solutions when $D=3$. However, as the delay bound increases the Heuristic BSG that is based on the best solution group performs better than the Heuristic BNC that iteratively determines the number of cells in each paging area. When $D=5$, the Heuristic BSG outperforms the BNC in seven cases out of ten.

The algorithm by Wang shows as good performance as the Heuristic BNC when $D=2$. When the delay bound is increased the solution gap from the Heuristic BNC is increased. Notice in <Table 1> that in every problem the expected paging cost of BSG is lower than or equal to that of the BNC. Also, the cost of BNC is lower than or equal to the cost of the procedure by Wang. This seems to be mainly due to that the Heuristic BNC Partitions the cells that satisfy the Lemma 3 and Corollary 1 of Section II-A.

Almost same trend is found in problems with $D=5$ and $D=100$. However, the solution gapamong the procedures is increased as the delay

〈Table 1〉 Partition of 30 cells into *D* paging areas

Problem Number	Algorithm	D=3			D=4			D=5		
		Solution	E[C]	E[D]	Solution	E[C]	E[D]	Solution	E[C]	E[D]
1	Optimal	(1,2,27)	1.23	1.10	(1,1,2,26)	1.12	1.11	(1,1,1,2,25)	1.11	1.11
	BSG	(2,2,26)	2.02	1.01	(1,2,2,25)	1.20	1.10	(1,1,2,2,24)	1.12	1.11
	BNC	(2,2,26)	2.02	1.01	(1,2,2,25)	1.20	1.10	(1,1,2,2,24)	1.12	1.11
	Wang	(2,2,26)	2.02	1.01	(2,2,2,24)	2.02	1.01	(2,2,2,2,22)	2.02	1.01
2	Optimal	(1,2,27)	1.62	1.21	(1,1,2,26)	1.32	1.24	(1,1,1,2,25)	1.26	1.25
	BSG	(2,2,26)	2.12	1.04	(1,2,2,25)	1.42	1.21	(1,1,2,2,24)	1.28	1.24
	BNC	(2,2,26)	2.12	1.04	(2,2,2,24)	2.08	1.04	(2,2,2,2,22)	2.08	1.04
	Wang	(2,2,26)	2.12	1.04	(2,2,2,24)	2.08	1.04	(2,2,2,2,22)	2.08	1.04
3	Optimal	(1,3,26)	2.11	1.31	(1,1,3,25)	1.63	1.39	(1,1,1,3,24)	1.49	1.42
	BSG	(2,3,25)	2.33	1.09	(1,2,3,24)	1.70	1.33	(1,1,2,3,23)	1.51	1.40
	BNC	(2,3,25)	2.33	1.09	(2,2,3,23)	2.20	1.10	(2,2,2,3,21)	2.20	1.10
	Wang	(3,3,24)	3.10	1.03	(2,3,3,22)	2.28	1.09	(2,2,3,3,20)	2.21	1.09
4	Optimal	(2,3,25)	2.74	1.17	(1,2,3,24)	2.09	1.47	(1,1,2,3,23)	1.83	1.63
	BSG	(3,3,24)	3.29	1.07	(1,3,3,23)	2.31	1.43	(1,1,3,3,22)	1.93	1.57
	BNC	(3,3,24)	3.29	1.07	(2,3,3,22)	2.53	1.17	(2,2,3,3,20)	2.40	1.19
	Wang	(3,3,24)	3.29	1.07	(3,3,3,21)	3.21	1.07	(3,3,3,3,18)	3.21	1.07
5	Optimal	(2,4,24)	3.38	1.27	(1,2,4,23)	2.68	1.63	(1,1,2,4,22)	2.34	1.89
	BSG	(3,4,23)	3.68	1.13	(2,3,4,21)	2.92	1.28	(1,2,3,4,20)	2.46	1.64
	BNC	(3,4,23)	3.68	1.13	(3,3,4,20)	3.46	1.14	(2,3,3,4,18)	2.86	1.29
	Wang	(4,4,22)	4.34	1.07	(3,4,4,19)	3.54	1.13	(3,3,4,4,16)	3.44	1.14
6	Optimal	(2,5,23)	4.44	1.39	(2,2,5,21)	3.58	1.50	(1,2,2,5,20)	3.14	2.20
	BSG	(4,5,21)	4.86	1.14	(2,4,5,19)	3.74	1.41	(1,2,4,5,18)	3.24	1.85
	BNC	(4,5,21)	4.86	1.14	(3,4,5,18)	4.04	1.25	(3,3,3,4,17)	3.85	1.27
	Wang	(4,5,21)	4.86	1.14	(4,4,5,17)	4.62	1.14	(3,4,4,4,15)	4.00	1.25
7	Optimal	(3,6,21)	5.90	1.38	(2,3,6,19)	4.85	1.68	(2,2,3,6,17)	4.36	2.59
	BSG	(5,6,19)	6.38	1.19	(3,4,6,17)	5.03	1.43	(2,3,4,5,16)	4.45	1.71
	BNC	(5,6,19)	6.38	1.19	(4,4,5,17)	5.41	1.31	(3,4,4,5,14)	4.85	1.45
	Wang	(5,6,19)	6.38	1.19	(4,5,5,16)	5.51	1.29	(4,4,5,5,12)	5.14	1.31
8	Optimal	(4,7,19)	8.47	1.49	(3,4,7,16)	7.19	1.76	(2,3,4,6,15)	6.53	3.06
	BSG	(5,7,18)	8.50	1.39	(4,5,6,15)	7.35	1.58	(2,3,4,6,15)	6.53	2.13
	BNC	(5,7,18)	8.50	1.39	(4,5,6,15)	7.35	1.58	(3,4,4,6,13)	6.67	1.83
	Wang	(6,7,17)	8.74	1.32	(5,6,6,13)	7.75	1.43	(4,5,5,5,11)	7.07	1.60
9	Optimal	(6,9,15)	13.16	1.68	(4,6,8,12)	11.75	2.07	(3,4,5,7,11)	10.96	3.66
	BSG	(7,9,14)	13.19	1.60	(5,6,8,11)	11.77	1.95	(4,4,5,7,10)	10.96	2.35
	BNC	(7,9,14)	13.19	1.60	(5,6,8,11)	11.77	1.95	(4,4,5,7,10)	10.96	2.35
	Wang	(8,8,14)	13.31	1.55	(6,7,7,10)	11.95	1.81	(5,6,6,6,7)	11.25	2.03
10	Optimal	(8,9,13)	15.07	1.70	(6,6,8,10)	13.58	2.08	(4,5,5,6,10)	12.77	3.97
	BSG	(9,9,12)	15.11	1.63	(5,6,8,11)	13.60	2.20	(5,5,5,6,9)	12.78	2.45
	BNC	(9,9,12)	15.11	1.63	(5,6,8,11)	13.60	2.20	(5,5,5,6,9)	12.78	2.45
	Wang	(9,9,12)	15.11	1.63	(7,7,6,10)	13.68	1.95	(6,5,6,5,8)	12.90	2.26

〈Table 2〉 Partition of 50 cells into *D* paging areas

Problem Number	Algorithm	D=3			D=4			D=5		
		Solution	E[C]	E[D]	Solution	E[C]	E[D]	Solution	E[C]	E[D]
1	Optimal	(1,2,47)	1.25	1.10	(1,1,2,46)	1.12	1.11	(1,1,1,2,45)	1.11	1.11
	BSG	(2,2,46)	2.02	1.01	(1,2,2,45)	1.20	1.10	(1,1,2,2,44)	1.12	1.11
	BNC	(2,2,46)	2.02	1.01	(2,2,2,44)	2.02	1.01	(2,2,2,2,42)	2.02	1.01
	Wang	(2,2,46)	2.02	1.01	(2,2,2,44)	2.02	1.01	(2,2,2,2,42)	2.02	1.01
2	Optimal	(1,3,46)	1.67	1.20	(1,1,3,45)	1.33	1.24	(1,1,1,3,44)	1.27	1.25
	BSG	(2,3,45)	2.13	1.04	(1,2,3,44)	1.43	1.21	(1,1,2,3,43)	1.29	1.24
	BNC	(2,3,45)	2.13	1.04	(2,2,3,43)	2.09	1.04	(2,2,2,3,41)	2.08	1.04
	Wang	(3,3,44)	3.03	1.00	(2,3,3,42)	2.12	1.04	(2,2,3,3,40)	2.85	1.04
3	Optimal	(1,3,46)	2.27	1.31	(1,1,3,45)	1.68	1.39	(1,1,1,3,44)	1.50	1.42
	BSG	(3,3,44)	3.11	1.03	(1,3,3,43)	1.93	1.31	(1,1,3,3,42)	1.58	1.39
	BNC	(3,3,44)	3.11	1.03	(3,3,3,41)	3.08	1.02	(2,2,3,3,40)	2.21	1.10
	Wang	(3,3,44)	3.11	1.03	(3,3,3,41)	3.08	1.02	(3,3,3,3,38)	3.08	1.03
4	Optimal	(2,4,44)	2.82	1.16	(1,2,4,43)	2.13	1.47	(1,1,2,4,42)	1.85	1.63
	BSG	(3,4,43)	3.33	1.07	(2,3,4,41)	2.53	1.17	(1,2,3,4,40)	2.01	1.47
	BNC	(3,4,43)	3.33	1.07	(3,3,4,40)	3.21	1.07	(3,3,3,4,37)	3.21	1.07
	Wang	(4,4,42)	4.13	1.03	(3,4,4,39)	3.26	1.07	(3,3,4,4,36)	3.21	1.07
5	Optimal	(2,5,43)	3.59	1.26	(1,2,5,42)	2.79	1.63	(1,1,2,5,41)	2.39	1.88
	BSG	(4,5,41)	4.39	1.06	(2,4,5,39)	3.10	1.27	(1,2,4,5,38)	2.55	1.63
	BNC	(4,5,41)	4.39	1.06	(3,4,5,38)	3.55	1.13	(3,3,4,5,35)	3.44	1.14
	Wang	(4,5,41)	4.39	1.06	(4,4,5,37)	4.27	1.07	(4,4,4,5,33)	4.27	1.07
6	Optimal	(3,6,41)	4.71	1.23	(2,3,6,39)	3.69	1.44	(1,2,3,6,38)	3.21	2.19
	BSG	(5,6,39)	5.61	1.08	(2,5,6,37)	4.02	1.39	(2,2,5,6,35)	3.44	1.50
	BNC	(5,6,39)	5.61	1.08	(4,5,6,35)	4.72	1.14	(3,4,4,6,33)	4.00	1.25
	Wang	(5,6,39)	5.61	1.08	(5,5,6,34)	5.43	1.08	(4,5,5,6,30)	4.70	1.14
7	Optimal	(4,8,38)	6.45	1.25	(2,4,7,37)	5.14	1.62	(2,2,4,7,35)	4.51	2.56
	BSG	(6,7,37)	7.18	1.13	(3,6,7,34)	5.45	1.39	(2,3,6,7,32)	4.67	1.68
	BNC	(6,7,37)	7.18	1.13	(5,6,7,32)	6.20	1.19	(4,5,6,7,28)	5.49	1.29
	Wang	(7,7,36)	7.82	1.09	(6,6,7,31)	6.84	1.13	(5,6,6,7,26)	6.15	1.19
8	Optimal	(5,10,15)	9.51	1.36	(3,5,9,33)	7.81	1.70	(3,3,5,9,30)	6.97	3.02
	BSG	(7,10,33)	9.84	1.23	(4,7,9,30)	7.99	1.51	(3,4,6,9,28)	7.00	1.78
	BNC	(7,10,33)	9.84	1.23	(6,7,9,28)	8.54	1.32	(4,5,6,9,26)	7.29	1.58
	Wang	(8,9,33)	10.25	1.19	(7,8,9,26)	9.12	1.25	(6,7,8,8,21)	8.38	1.33
9	Optimal	(8,13,29)	16.60	1.53	(6,8,12,24)	14.36	1.81	(4,6,7,11,22)	13.17	3.62
	BSG	(9,13,28)	16.60	1.48	(6,9,12,23)	14.40	1.78	(5,6,8,11,20)	13.20	2.07
	BNC	(9,13,28)	16.60	1.48	(7,8,12,23)	14.45	1.73	(7,8,8,10,17)	13.70	1.79
	Wang	(11,13,26)	16.98	1.39	(9,10,11,20)	15.03	1.55	(8,8,9,9,16)	14.02	1.70
10	Optimal	(13,14,23)	24.82	1.72	(9,10,13,18)	22.40	2.14	(8,8,9,10,15)	21.02	4.06
	BSG	(13,14,23)	24.82	1.72	(9,10,13,18)	22.40	2.14	(8,8,9,10,15)	21.02	2.45
	BNC	(13,14,23)	24.82	1.72	(10,11,12,17)	22.44	2.05	(8,8,9,10,15)	21.02	2.45
	Wang	(15,14,21)	24.95	1.63	(11,12,11,16)	22.57	1.97	(10,9,9,9,13)	21.26	2.22

〈Table 3〉 Partition of 100 cells into *D* paging areas

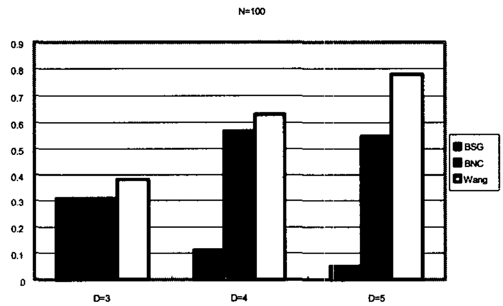
Problem Number	Algorithm	D=3			D=4			D=5		
		Solution	E[C]	E[D]	Solution	E[C]	E[D]	Solution	E[C]	E[D]
1	Optimal	(1,2,97)	1.30	1.01	(1,1,2,96)	1.13	1.11	(1,1,1,2,95)	1.11	1.11
	BSG	(2,2,96)	2.03	1.01	(1,2,2,95)	1.20	1.10	(1,1,2,2,94)	1.12	1.11
	BNC	(2,2,96)	2.03	1.01	(2,2,2,94)	2.02	1.01	(2,2,2,2,92)	2.02	1.01
	Wang	(2,2,96)	2.03	1.01	(2,2,2,94)	2.02	1.01	(2,2,2,2,92)	2.02	1.01
2	Optimal	(1,3,96)	1.75	1.20	(1,1,3,95)	1.35	1.24	(1,1,1,3,94)	1.27	1.25
	BSG	(3,3,94)	3.03	1.00	(1,3,3,93)	1.61	1.20	(1,1,3,3,92)	1.32	1.24
	BNC	(3,3,94)	3.03	1.00	(3,3,3,91)	3.02	1.00	(2,3,3,3,89)	2.12	1.04
	Wang	(3,3,94)	3.03	1.00	(3,3,3,91)	3.02	1.00	(3,3,3,3,88)	3.02	1.00
3	Optimal	(2,4,94)	2.43	1.09	(1,2,4,93)	1.73	1.33	(1,1,1,4,93)	1.52	1.42
	BSG	(3,4,93)	3.13	1.03	(1,3,4,92)	1.94	1.31	(1,1,3,4,91)	1.58	1.39
	BNC	(3,4,93)	3.13	1.03	(3,3,4,90)	3.08	1.03	(3,3,3,4,87)	3.08	1.03
	Wang	(4,4,92)	4.04	1.00	(3,4,4,89)	3.11	1.03	(3,4,4,4,85)	3.11	1.03
4	Optimal	(2,5,93)	2.95	1.16	(1,2,5,92)	2.18	1.46	(1,1,2,5,91)	1.87	1.63
	BSG	(4,5,91)	4.15	1.03	(2,4,5,89)	2.66	1.16	(1,2,4,5,88)	2.07	1.47
	BNC	(4,5,91)	4.15	1.03	(4,4,5,87)	4.11	1.03	(3,4,4,5,84)	3.26	1.07
	Wang	(4,5,91)	4.15	1.03	(4,5,5,86)	4.13	1.03	(4,4,5,5,82)	4.11	1.03
5	Optimal	(2,6,92)	3.86	1.25	(1,2,6,91)	2.93	1.63	(1,1,2,6,90)	2.46	1.88
	BSG	(5,6,89)	5.23	1.03	(2,5,6,87)	3.31	1.26	(1,2,5,6,86)	2.65	1.63
	BNC	(5,6,89)	5.23	1.03	(5,5,6,84)	5.16	1.03	(4,4,5,6,81)	4.27	1.07
	Wang	(5,6,89)	5.23	1.03	(5,6,6,83)	5.19	1.03	(5,5,6,6,78)	5.16	1.03
6	Optimal	(3,8,89)	5.05	1.22	(2,3,8,87)	3.82	1.44	(1,2,3,7,87)	3.29	2.18
	BSG	(6,7,87)	6.44	1.05	(3,6,7,84)	4.39	1.23	(2,3,6,7,82)	3.58	1.44
	BNC	(6,7,87)	6.44	1.05	(5,6,7,82)	5.50	1.08	(5,5,6,7,77)	5.43	1.08
	Wang	(7,7,86)	7.26	1.03	(6,7,7,80)	6.34	1.05	(6,6,7,7,74)	6.30	1.05
7	Optimal	(4,10,86)	6.98	1.25	(3,4,10,83)	5.39	1.43	(2,3,4,10,81)	4.64	2.55
	BSG	(8,10,82)	8.71	1.06	(4,8,9,79)	6.09	1.25	(2,4,8,9,77)	4.98	1.61
	BNC	(8,10,82)	8.71	1.06	(7,8,9,76)	7.72	1.09	(6,7,8,9,70)	6.91	1.13
	Wang	(9,10,81)	9.50	1.04	(8,9,9,74)	8.55	1.06	(7,8,9,9,67)	7.70	1.09
8	Optimal	(6,13,81)	10.58	1.28	(4,6,13,77)	8.31	1.52	(3,4,6,13,74)	7.24	2.99
	BSG	(11,13,76)	12.48	1.09	(5,10,13,72)	8.87	1.36	(4,5,10,13,68)	7.63	1.56
	BNC	(11,13,76)	12.48	1.09	(9,10,13,68)	10.58	1.15	(7,8,10,12,63)	9.09	1.25
	Wang	(12,13,75)	13.17	1.07	(10,11,12,67)	11.33	1.12	(9,10,11,12,58)	10.52	1.15
9	Optimal	(11,20,69)	19.91	1.35	(7,11,19,63)	16.39	1.65	(5,7,10,19,59)	14.61	3.53
	BSG	(15,20,65)	20.74	1.23	(9,14,19,58)	16.80	1.49	(6,9,14,18,53)	14.89	1.79
	BNC	(15,20,65)	20.74	1.23	(11,14,18,57)	17.30	1.40	(9,11,13,17,50)	15.62	1.54
	Wang	(17,19,64)	21.61	1.19	(15,16,18,51)	19.27	1.25	(13,14,15,16,42)	17.71	1.33
10	Optimal	(27,31,42)	52.37	1.75	(20,22,25,33)	47.55	2.14	(15,16,18,22,29)	44.78	4.22
	BSG	(27,31,42)	52.37	1.75	(20,21,25,34)	47.55	2.16	(15,17,19,21,28)	44.80	2.58
	BNC	(27,31,42)	52.37	1.75	(20,21,25,34)	47.55	2.16	(17,18,18,20,.27)	44.87	2.48
	Wang	(30,30,40)	52.49	1.69	(24,23,23,30)	47.79	2.00	(19,19,19,19,24)	45.09	2.36

〈Table 4〉 Gap from the Paging Cost by Algorithm OPA

# of Cells	Gap	D=3			D=4			D=5		
		BSG	BNC	Wang	BSG	BNC	Wang	BSG	BNC	Wang
N=30	Min	0.002	0.002	0.003	0.001	0.001	0.007	0.000	0.000	0.010
	Max	0.642	0.642	0.642	0.105	0.576	0.804	0.055	0.651	0.820
	Avg	0.153	0.153	0.213	0.049	0.177	0.316	0.020	0.203	0.375
N=50	Min	0.000	0.000	0.005	0.000	0.002	0.008	0.000	0.000	0.011
	Max	0.616	0.616	0.814	0.188	0.833	0.833	0.086	0.820	1.244
	Avg	0.200	0.200	0.300	0.077	0.357	0.432	0.035	0.366	0.575
N=100	Min	0.000	0.000	0.002	0.000	0.000	0.005	0.000	0.002	0.007
	Max	0.731	0.731	0.731	0.220	1.237	1.237	0.107	1.026	1.378
	Avg	0.309	0.309	0.385	0.110	0.565	0.628	0.051	0.546	0.779

bound and the number of cells increased. More specifically, the gap between the Heuristic BSG and the Algorithm OCP is reduced as the delay bound is increased. The BSG well converges to the optimal solution as the problem size is increased. However, the gap between the procedure by Wang and the Algorithm OCP is increased as the delay bound increases.

<Table 4> shows the performance of each procedure compared to the optimal solution by Algorithm OCP. As shown in the table when $D=5$ the average gap of the proposed Heuristic BSG from the optimal solution is 2~5% while those of the method by Wang is 38~78% depending on the number of cells. Clearly, the proposed two heuristics outperform the algorithm by Wang et al. The gain by the heuristic BSG compared to the algorithm by Wang is 6~69% in problems with $D=100$. The average gaps by the three procedures are compared as in <Figure 3>. It shows that the proposed BSG performs better than two other procedures as the delay bound increases.



〈Figure 3〉 Gap from the optimal solution with N=100

5. Conclusion

The sequential paging problem with delay bound is investigated by examining the necessary conditions for the optimal cell partitioning. In addition to the well-known condition of non-increasing order of cell probabilities, the condition of nondecreasing order of number of cells is developed. The optimal cell partitioning OCP and two heuristics BSG and BNC are proposed based on the necessary conditions and the fathoming rule that efficiently cuts out unnecessary solution space and expedites the search process.

The performance of the proposed sequential algorithms are experimented and compared with existing algorithm. Computational results show that the proposed Heuristic BSG well converges to the optimal solutions even if the problem size increases. However, the Heuristic BNC and the best-known existing algorithm tend to diverge from the optimal solution as the number of cells and the delay bound increase. The solution gap of the Heuristic BSG from the optimal solution records 2~5% while that of the best-known existing algorithm records 38~78% when the delay bound is five.

Reference

[1] Abutaleb, A. and V.O.K. Li, "Paging strategy in personal communication systems," *ACM-Baltzer Journal of Wireless Networks* (WINET), Vol.3(1997), pp.195-204.

[2] Akyildiz, I.F., J. McNair., J.S.M. Ho., H. Uzunalioğlu, and W. Wang, "Mobility Management in Next-Generation Wireless Systems," *Proceedings of the IEEE*, Vol.87(1999), pp.1347-1348.

[3] Akyildiz, I.F., H.S.M. Ho, and Y.B. Lin, "Movement-Based Location Update and Selective Paging for wireless Networks," *IEEE/ACM Transactions on Networking*, Vol.4(1996), pp.629-638.

[4] Bar-Noy, A., I. Kessler, and M. Sidi, "Mobile Users : To Update or not to Update?," *IEEE INFOCOM' 94*, (1994), pp.570-576.

[5] Bhaskar Krishnamachari, Rung-Hung Gau, Stephen B. Wicker and Zygmunt J. Haas, "On the Performance of Sequential Paging for Mobile User Location," *IEEE VTC fall 2001*.

[6] Goodman, D., P. Krishnan, and B. Sugla, "Minimization Queueing Delays and Number of Messages in Mobile Phone Location," *Journal on Mobile Networks and Applications*, Vol.1, No.1(1996), pp.39-48.

[7] Lee, H.C. and J. Sun, "Mobile Location Tracking by Optimal Paging Zone Partitioning," *IEEE 6th International Conference on*, vol.1(1997), pp.168-172.

[8] Madhavapeddy, S., K. Basu, and A. Roberts, "Adaptive Paging Algorithms for Cellular Systems," Wireless Information Networks: Architesture, Resource Management and Mobile Data, (1996), pp.83-101.

[9] Rose, C. and R. Yates, "Minimizing the Average Cost of Paging Under Delay Constraints," *ACM-Baltzer Journal of Wireless Networks* (WINET), Vol.1(1995), pp.211-219.

[10] Wang, W., I.F. Akyildiz, and G.L. Stűber, "An Optimal Paging Scheme for Minimizing Signaling Costs Under Delay Bounds," *IEEE Communication Letters*, Vol.5, No.2 (2001), pp.43-45.

[11] Wang, W., I.F. Akyildiz, and G.L. Stűber, "Effective Paging Schemes with Delay Bounds as QoS Constraints in Wireless Systems," *ACM-Baltzer Journal of Wireless Networks* (WINET), Vol.7(2001), pp.455-466.

[12] Wong, V.W-S. and V.C.M. Leung, "Location Management for Next-Generation Personal Communications Networks," *IEEE Network*, 14(2000), pp.18-24.