

# A Case Study of Belady's Anomaly and the Binomial Distribution

**Kirby McMaster**  
**St. Mary's College of**  
**Maryland, Piney**  
**Point, MD, USA**

**Samuel Sambasivam**  
**Azusa Pacific**  
**University, Azusa,**  
**CA, USA**

**Nicole Anderson**  
**Winona State**  
**University, Winona,**  
**MN, USA**

[kmcmaster@weber.edu](mailto:kmcmaster@weber.edu)

[ssambasivam@apu.edu](mailto:ssambasivam@apu.edu)

[nanderson@winona.edu](mailto:nanderson@winona.edu)

## Abstract

In Operating Systems courses, instructors often assign case studies that involve programming to enhance course objectives. One topic that usually arouses students' interest is Belady's anomaly. In virtual memory systems with demand paging, the page fault rate of a process varies with the number of memory frames allocated to the process. When an increase in allocated frames leads to an increase in page faults, *Belady's anomaly* is said to occur. In this paper, we present a binomial probability model for Belady's anomaly. In this case study, computer simulation was used to estimate the parameters of the model over a design region of process sizes and reference string lengths. The number of unshared pages in  $K$  vs.  $K+1$  memory frames was determined and was used to compute the parameters  $N$  and  $P$  of a corresponding binomial distribution. A chi-square goodness-of-fit test was performed that related the binomial model to the occurrence of Belady's anomaly in the simulation results.

**Keywords:** Belady's anomaly, virtual memory, demand paging, page replacement, binomial.

## Introduction

In Operating Systems courses, instructors often assign case studies that involve programming to enhance course objectives (Lions, 1978; Keromytis, et al, 2006). One topic that typically arouses students' interest is Belady's anomaly. In virtual memory systems with demand paging (Randall & Kuehner, 1968), the page fault rate of a process varies with the number of frames available for the process. When an increase in allocated frames leads to an increase in page faults, *Belady's anomaly* is said to occur (Belady, 1966; Belady, Nelson, & Shedler 1969).

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [0HPublisher@InformingScience.org](mailto:0HPublisher@InformingScience.org) to request redistribution permission.

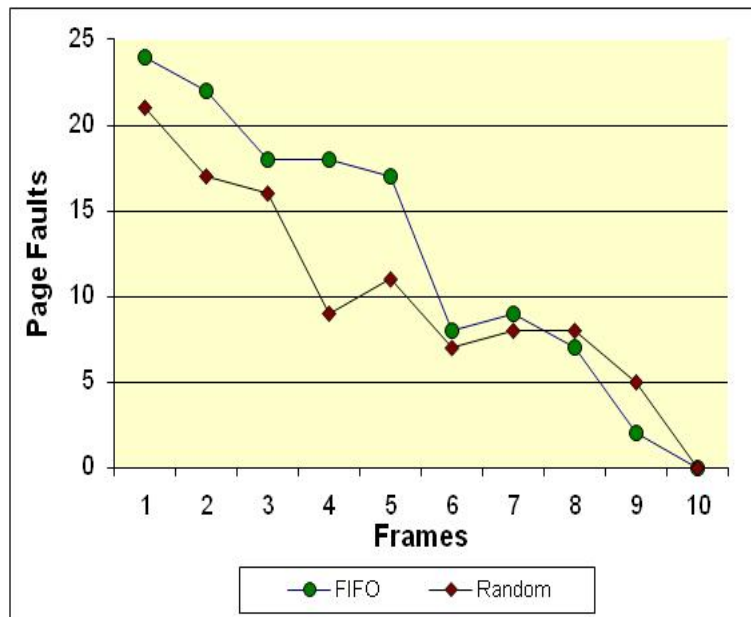
It is known that the occurrence rate for Belady's anomaly depends on which page replacement algorithm is implemented. Examples of Belady's anomaly for the First-In, First-Out (FIFO) algorithm are presented in several Operating Systems textbooks (Deitel and Choffnes, 2004; Dhamdhere, 2008; Schlesinger & Garrido, 2007; Silberschatz, Galvin, and Gagne, 2008; Stuart, 2008; Warford, 2009). On the other hand, Belady's

## Belady's Anomaly

anomaly cannot occur when page replacement is based on a stack algorithm (Mattson, Gecsei, Sultz, & Traiger, & 1970).

Examples of Belady's anomaly for the FIFO and Random Page algorithms are displayed as line graphs in Figure 1. In these two cases, the process size is 10 pages, and the reference string length is 25. The reference strings are not shown, but were selected from the random reference strings generated in our simulation study. In this simulation, each set of  $K$  frames was "pre-filled" so that its pages initially were a subset of the pages in  $K+1$  frames.

Normally, as more memory becomes available, the page fault rate drops. In the FIFO graph, when the number of frames allocated moves from 6 to 7, the number of page faults increases from 8 to 9 (an anomalous *bump* in the graph). The Random Page graph displays three anomaly bumps, when the frame level goes from 4 to 5, from 6 to 7, and from 7 to 8.

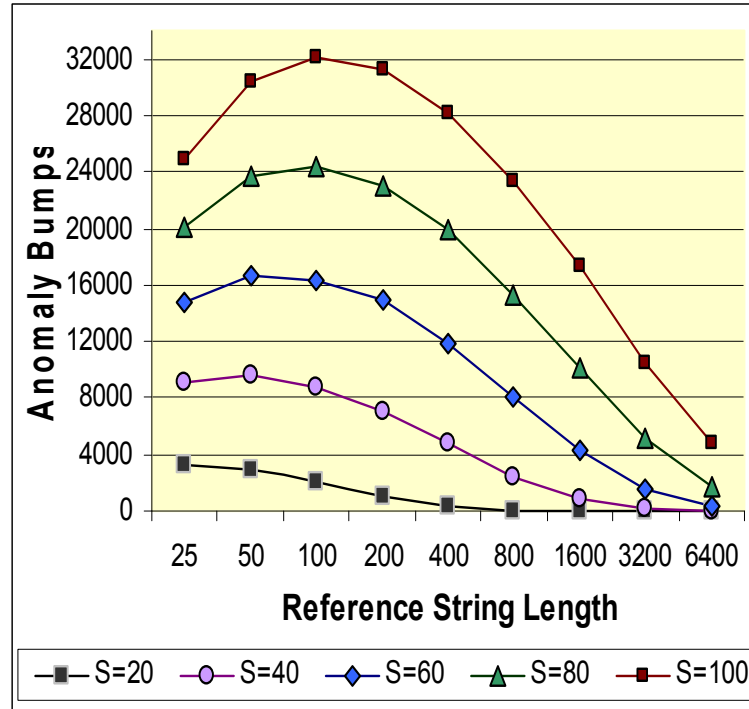


**Figure 1. Belady's Anomaly Bumps.**

Process size = 10, and reference string length = 25.

In an earlier study (McMaster, Sambasivam, & Anderson, 2009a), we used computer simulation to analyze conditions that affect how often Belady's anomaly occurs for the FIFO and Random Page algorithms. For each process size  $S$ , reference string length  $L$ , and frame level  $K$  in the design region, we counted the total number of page fault increases, or anomaly *bumps*, for  $K$  vs.  $K+1$  frames that appeared in 1000 reference strings.

Figure 2 presents the anomaly bump counts for the Random Page algorithm. The bump counts are surprisingly large, occurring as often as 32152 times in 1000 reference strings. For each process size  $S$ , the maximum number of bumps occurs when the reference strings are relatively short (100 or less).



**Figure 2. Random Page Anomaly Bumps.**  
For each 1000 reference strings.

### Probability Model

In our previous study, we recorded how often Belady's anomaly occurred, but we did not discuss the reasons for the observed patterns. An explanation of why certain conditions influence Belady's anomaly requires an appropriate probability model (Acevedo, 1972; Franaszek & Wagner, 1974). In a recent second paper, we presented a preliminary probability model for Belady's anomaly (McMaster, Sambasivam, & Anderson, 2009b). The current paper presents a more detailed probability model for Belady's anomaly that approximates a binomial distribution.

Consider the memory regions for K frames vs. K+1 frames shown in Figure 3. At any point in time, each of the S process pages will be in one of four disjoint groups:

- Group 1: Unshared pages in K frames (A pages).
- Group 2: Unshared pages in K+1 frames (A+1 pages).
- Group 3: Shared pages in K frames and K+1 frames.
- Group 4: Not in memory.

Belady's anomaly occurs when, for a given reference string, more page references are in Group 1 than in Group 2. This leads to fewer page faults for K frames than K+1 frames.

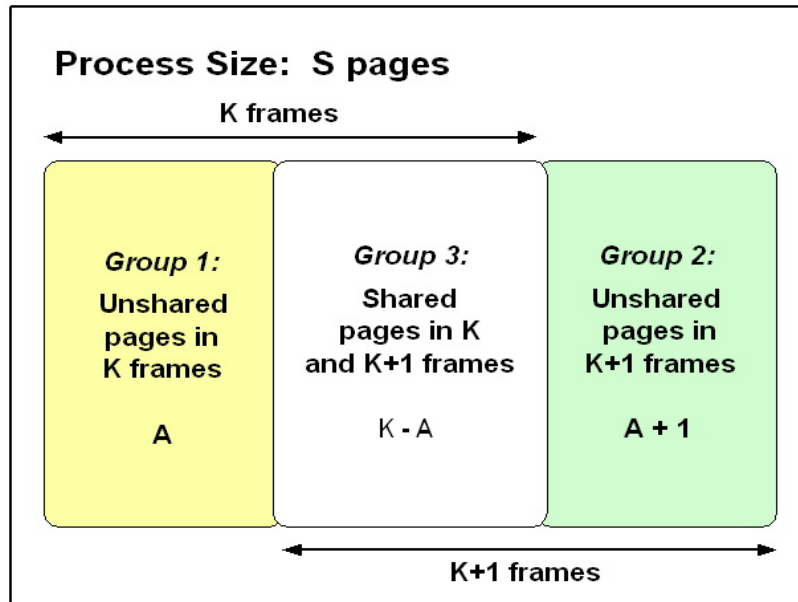


Figure 3. Memory Groups for Page References.

A probability model can be defined for random page references, based on the group in which a page might be located. The Bernoulli random variable  $X$  shown in Table 1 represents the result of a single page reference when that page is in Group 1 or Group 2. If the page reference is in Group 3 (no page faults for  $K$  or  $K+1$  frames) or Group 4 (page faults for both  $K$  and  $K+1$  frames), the value is ignored, since neither case affects Belady's anomaly.

Table 1. Bernoulli Random Variable for a Single Page Reference.

Group	X	Probability	Page Faults
1	1	$P = A/(2A+1)$	for $K+1$ frames
2	0	$1-P = (A+1)/(2A+1)$	for $K$ frames
3 or 4	--	Ignore page reference	no effect on Belady's anomaly

If page faults for a reference string of length  $L$  are accumulated using  $Y = \sum X$ , Belady's anomaly will occur when there are more page faults for  $K+1$  frames ( $X = 1$  values) than for  $K$  frames ( $X = 0$  values). In this case, the random variable  $Y$  will be greater than  $N/2$ , where  $N$  is the number of Group 1 and Group 2 pages in the reference string. Thus, the probability of an anomaly is  $P[Y > N/2]$  for given  $N$  and  $K$ . We note that  $N$  varies randomly and is usually much smaller than  $L$ . If  $A$  is constant, the expected value of  $N$  is  $L(2A+1)/S$ .

Our preliminary model for Belady's anomaly is that the random variable  $Y$  has approximately a binomial distribution with parameters  $N$  and  $P$ . The binomial distribution (Johnson, Kotz, & Balakrishnan 2005) is based on the following assumptions:

1. The number of *trials*  $N$  (page references in Group 1 or Group 2) is fixed.
2. The probability  $P$  of a *success* ( $X = 1$ ) is constant.
3. The trials are independent.

Our simulation results show that these assumptions are not entirely satisfied, but it is insightful to use the binomial model as an approximation. In this paper, we will evaluate the suitability of this model.

If  $Y$  has a binomial distribution with parameters  $N$  and  $P$ , and  $P[Y > N/2]$  represents the probability of an anomaly for  $K$  vs.  $K+1$  frames, then Belady's anomaly depends on the following factors:

1. A longer reference string length  $L$  *decreases* the likelihood of an anomaly (larger  $N$ , and  $P = A/(2A+1) < 0.5$ ).
2. A larger process size  $S$  *increases* the likelihood of an anomaly (smaller  $N$ ).
3. A greater number of unshared pages  $A$  both *increases* and *decreases* the likelihood of an anomaly (larger  $P$ , but also larger  $N$ ).

Additional considerations that affect Belady's anomaly include:

4. The value of  $A$  depends on process size  $S$  and frame level  $K$ . The pattern of these relationships and their effect on Belady's anomaly are not straightforward.
5. Larger  $S$  means that there are more frame pairs  $K$  vs.  $K+1$  in which an anomaly can appear. This increases the likelihood of an anomaly.
6. The value of  $A$  is *not constant*. Initially, the value of  $A$  is 0 while the  $K$  frames and  $K+1$  frames are being filled. Additional page references gradually increase the value of  $A$  up to an equilibrium value.
7. The amount of change in  $A$  depends on the page replacement algorithm. With a stack algorithm,  $A$  is always 0, so Belady's anomaly never occurs. The number of unshared pages increases for Random Page and FIFO, but at different rates and with different equilibrium values.

## Methodology

This study examines Belady's anomaly for the Random Page algorithm. The FIFO page replacement algorithm is left as a programming project for Operating Systems students.

Computer simulation (Fine, Jackson, & McIsaac, 1966) was performed using a program written specifically for this research. In our design, we varied three conditions that affect the occurrence of Belady's anomaly:

1. Process size  $S$  in pages.
2. Reference string length  $L$ .
3. Memory frames  $K$  allocated to the process.

In the simulation runs, the process size ranged from 20 to 100 pages in increments of 20. The length of the reference string varied from 25 to 6400 page references, with each value being twice the preceding value. In each simulation run, one process size and one reference string length were selected. Then 1000 random reference strings were generated using a uniform random number generator. For each reference string, the number of page faults was computed for each memory allocation from 1 frame up to the process size.

For each memory allocation, frames were "pre-filled" before any reference string page faults were counted. Specifically,  $K$  memory frames were filled with page numbers  $K$  (first out) through 1 (last in). Similarly,  $K+1$  frames were filled with page numbers  $K+1$  (first out) through 1 (last in). These allocations insured that initially the pages in  $K$  frames were a subset of the pages in  $K+1$  frames, and that the pages were in the same order in the two queues. Our computer simulation

calculated the number of unshared pages A for each pair of frame levels (K vs. K+1) under each combination of design conditions.

## Unshared Page Patterns

In this section, we examine relationships between process size S, reference string length L, allocated frames K, and unshared pages A for the Random Page algorithm. We initially fix K as 50% of S and focus on how A depends on S and L. After determining equilibrium values for A, we observe the pattern of these equilibrium values with S and K.

### Unshared Pages and Reference Strings

Figure 4 describes how A varies as the reference string length L increases up to 6400 for process sizes S between 20 and 100, when the number of allocated frames K is 50% of S. In this case, 50% represents K = 10 frames for 20 pages, K = 20 frames for 40 pages, etc.

The initial value of A for length L = 0 is 0 because of how frames are pre-filled in the simulation. As L increases beyond 100, the number of unshared pages approaches an equilibrium value that depends on the process size. For L = 200 or greater, the slope of the line is essentially 0 for each process size.

The largest value for A in Figure 4 is 16.1 unshared pages in K = 50 frames (vs. K+1 frames) for process size S = 100. In this case, K-A = 33.9 pages are shared with the K+1 frame allocations. This equilibrium A value is reached when the reference string length L is 200.

By comparison, the maximum value for A when S = 20 is 2.8 unshared pages in K = 10 frames. Here, K-A = 7.2 pages are shared with the K+1 frame contents. This equilibrium A value is reached before L reaches 100. With Random Page, the initial pages in memory are "flushed out" fairly rapidly.

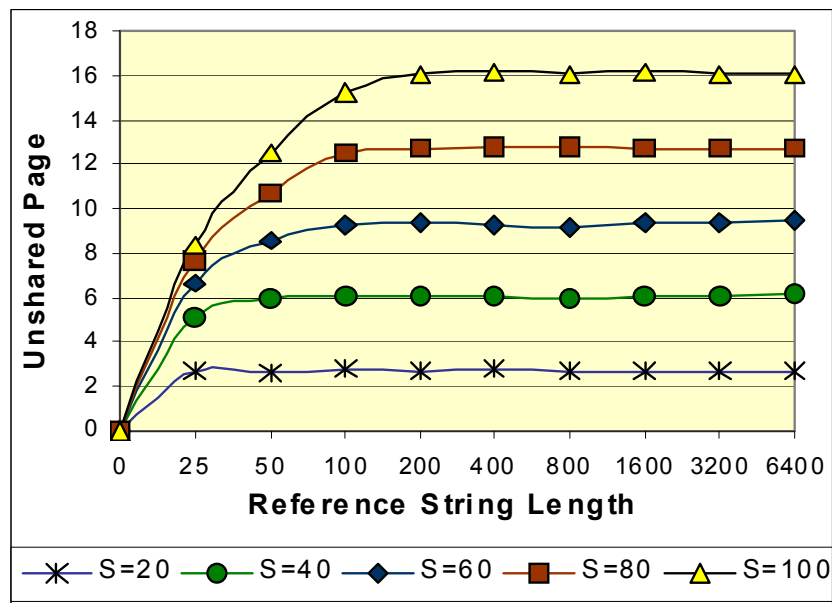


Figure 4. Unshared Pages for K vs. K+1 Frames. By process size S and reference string length L. Frame level K = 50% of S.

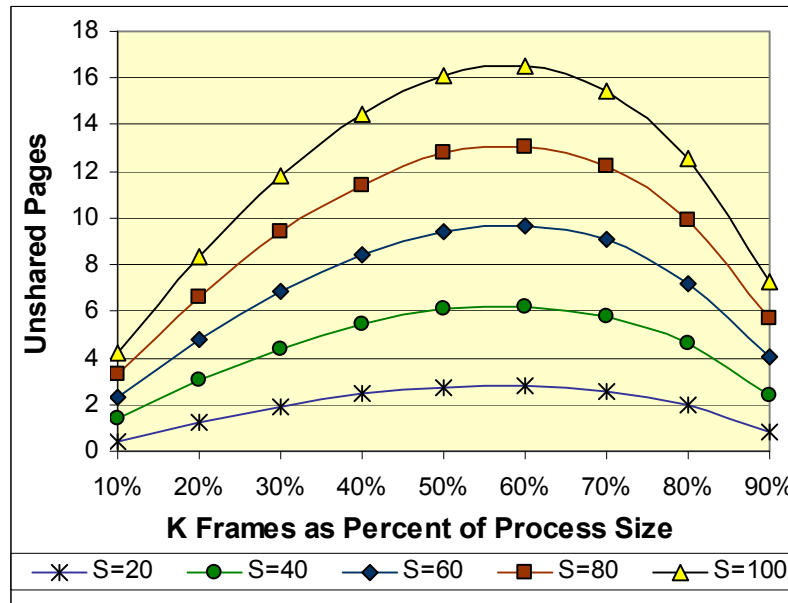
### Unshared Pages and Allocated Frames

We now describe how the number of unshared pages depends on allocated frames (expressed as a percent of S). Figure 5 summarizes the relationship of equilibrium A values to process size S (20 to 100) and allocated frames K (10% to 90% of S).

The Random Page pattern for equilibrium A values has some interesting features.

1. The only equilibrium A value less than 1.0 occurs when S is 20 and K is 10% of S (K = 2).
2. For each fixed process size S, as K grows from 10% to 90%, the value of A increases to a maximum, and then starts to decrease.
3. For each fixed S, the largest equilibrium A value occurs when K is 60% of S.
4. Larger process sizes have larger equilibrium A values.
5. The largest value of A in Figure 5 is 16.5 unshared pages for S = 100 and K = 60.

Few unshared pages occur under Random Page when the allocated frame level is near 0% or 100% of S. However, the number of unshared pages remains relatively large over a wide range of frame levels.



**Figure 5. Equilibrium Unshared Pages in K Frames.**  
By process size S and frame level K as percent of S.

### Unshared Pages and Anomalies

Table 2 lists, for each process size S, the maximum number of anomaly bumps. These values are taken from the data used to generate Figure 1. As S becomes larger, the maximum anomaly bump count grows substantially.

Table 2 includes, for each process size S, the equilibrium unshared pages A when the allocated frame level K is between 50% and 70% of S. This range of frame levels was chosen because they exhibit the largest A values for Random Page. As S becomes larger, the equilibrium values of A for each frame level increase linearly (correlation = 0.999). Table 2 also shows the ratio of A to S when K is 60% of S. The A values at this frame level, which range from 2.82 to 16.5, increase

very slightly when stated as a fraction of S. These results suggest that the effects of S and A on anomaly counts are confounded, making it difficult to estimate their contributions separately.

**Table 2. Anomaly Summary for Random Page.**  
Maximum anomaly bumps vs. equilibrium unshared pages.

<b>Process Size S</b>	<b>Max Bumps</b>	<b>Unshared Pages A K = 50%S</b>	<b>Unshared Pages A K = 60%S</b>	<b>Unshared Pages A K = 70%S</b>	<b>A/S K =60%S</b>
20	<b>3236</b>	2.76	<b>2.82</b>	2.58	<b>0.141</b>
40	<b>9547</b>	6.07	<b>6.23</b>	5.79	<b>0.156</b>
60	<b>16638</b>	9.38	<b>9.66</b>	9.06	<b>0.161</b>
80	<b>24400</b>	12.77	<b>13.08</b>	12.26	<b>0.164</b>
100	<b>32152</b>	16.11	<b>16.51</b>	15.45	<b>0.165</b>

Two issues limit the validity of this Table 2 comparison of unshared pages with anomalies. First, for each process size, the maximum number of anomalies occurs before A reaches its equilibrium value. Competing effects seem to be at work here. Anomalies increase as A becomes larger, but anomalies decrease when L grows longer. The highest anomaly bump frequencies involve a trade-off between these two effects.

The second issue is that the anomaly bump counts in Table 2 are the total of anomalies across all frame levels, whereas the unshared page values apply only to the 50% to 70% frame levels. A better comparison would be to match anomalies with unshared pages at specific frame levels.

### ***Unshared Pages and the Binomial Model***

We now use our simulation results to evaluate the proposed binomial probability model. We stated earlier that unshared pages A is related to binomial parameters P and N. Since A for Random Page grows rapidly at the start of each reference string, we concentrate our analysis on equilibrium values of A.

Table 3 presents the equilibrium A values for each process size S when allocated frames K is 60% of S. Each A value is used to estimate P as  $A/(2A+1)$ . The estimate for N is  $L(2A+1)/S$ , which indicates that N is proportional to reference string length L. Hence, in Table 3, we state N as a percent of L.



**Table 3. Equilibrium Unshared Pages and Binomial Model Parameters.** Frame level  $K = 60\%$  of  $S$ .

Process Size $S$	Unshared Pages $A$	$P = A/(2A+1)$	$N$ as % of $L$
20	2.82	<b>0.425</b>	<b>33.2%</b>
40	6.22	<b>0.463</b>	<b>33.6%</b>
60	9.67	<b>0.475</b>	<b>33.9%</b>
80	13.07	<b>0.482</b>	<b>33.9%</b>
100	16.52	<b>0.485</b>	<b>34.0%</b>

Even though equilibrium unshared pages  $A$  ranges from 2.82 to 16.52 (almost a 6X increase), the ranges for  $P$  and  $N$  are much smaller. The  $P$  values increase from 0.425 to 0.485, while  $N$  values increase by less than 1%. However, small differences in  $P$  values can have a moderately large effect on the probability of an anomaly, as will be shown in the next subsection.

### ***Binomial Model and Anomalies***

To evaluate our binomial model for Random Page anomalies, we need to compare observed anomalies with the number predicted by the model. Our simulation results include anomaly bump counts for many process sizes, reference string lengths, and allocated frame levels. As our test case, we chose a reference string length  $L$  of 400, since all  $A$  values reach equilibrium within the first 100 page references. We also fixed allocated frames  $K$  as 60% of process size  $S$ .

Table 4 summarizes the data used to perform a Chi Square goodness-of-fit test (Snedecor and Cochran, 1989) for the binomial model. For each process size  $S$ , the  $P$  and  $N$  values are estimated by the equilibrium  $A$  values listed in Table 3. The expected number of anomaly bumps was calculated using the upper-tail binomial probability  $P[Y > N/2]$ , multiplied by the number of random reference strings (1000 for each design condition). The observed bump frequencies were obtained from the detailed simulation output.

**Table 4. Binomial Model and Anomaly Bumps.** Reference string  $L = 400$ , and frames  $K = 60\%$  of  $S$ .

Process Size $S$	$P$	$N$	Exp Bumps	Obs Bumps
20	<b>0.425</b>	<b>133</b>	40.1	30
40	<b>0.463</b>	<b>134</b>	170.9	171
60	<b>0.475</b>	<b>136</b>	254.5	244
80	<b>0.482</b>	<b>136</b>	302.8	336
100	<b>0.485</b>	<b>136</b>	334.0	355

A Chi-Square statistic was computed from the five observed vs. expected frequency pairs, yielding a value of 7.92. The degrees of freedom (df) for this test is 5, since separate sets of reference

strings were generated at each design point. For a .05 significance level, the critical value for the test statistic is 11.07. Therefore, we fail to reject the binomial model as an explanation for the selected simulation anomaly bump counts.

## Summary and Conclusions

In this study we provided a binomial probability model to explain the occurrence of Belady's anomaly in terms of unshared pages for  $K$  vs.  $K+1$  memory frames. This model was applied to the Random Page replacement algorithm. We used computer simulation to estimate the number of unshared pages under three conditions: (1) process size, (2) reference string length, and (3) allocated frames. We then related the number of unshared pages to anomaly bump counts obtained in an earlier study.

For the Random Page algorithm, we found that for each process size  $S$  and frame allocation level  $K$ , the number of unshared pages  $A$  increased from 0 up to an equilibrium value as the reference string became longer. For larger process sizes, the  $A$  values became larger, but required slightly longer reference strings to reach equilibrium. Within the design region, maximum  $A$  values were attained when the reference string length was 200 or less.

The maximum equilibrium  $A$  value in the design region was 16.52, for  $S = 100$  and  $K = 60$ . The largest values of  $A$  for each process size typically occurred when  $K$  was 60% of  $S$ . Relatively large  $A$  values and anomaly bump counts occurred over a wide range of frame levels (10% through 90% of  $S$ ) for Random Page.

Belady's anomaly bumps are more frequent for large  $A$  values, especially when the large  $A$  values occur over a wide range of frame levels. Since long references strings decrease the probability of an anomaly bump, anomalies are more likely when equilibrium  $A$  values are reached quickly. These unshared page features combine to produce a high Belady's anomaly occurrence rate for Random Page.

A Chi Square goodness-of-fit test was performed to evaluate how well the binomial model explained the observed Random Page anomaly bump frequencies. The test compared observed vs. predicted bump frequencies for five points in the design region. The Chi Square statistic was below the .05 critical value, which does not allow us to reject the binomial model as a reasonable explanation for the tested simulation results.

## Research for Students

The case study in this paper focused on the occurrence rate of Belady's anomaly for the Random Page algorithm. The FIFO algorithm provides another case study and programming assignment, where Operating Systems students can extend these simulation results. Our preliminary findings for FIFO include:

1. Anomalies are much less frequent for FIFO than Random Page.
2. FIFO produces fewer unshared pages  $A$ .
3. The number of unshared pages  $A$  reaches equilibrium more slowly (requiring longer reference strings).
4. The largest equilibrium  $A$  values and anomaly rates occur when the allocated frame level is 70-80% of the process size.

Students can be asked to verify the above results, as well as search for additional patterns in their simulation data. During the case study, students should attempt to evaluate their findings both in

terms of the binomial probability model and based on the observed simulation behavior for the FIFO page replacement algorithm.

## References

- Acevedo, M. F. (1972). *A probabilistic study of two-level storage hierarchies*. M.S. Thesis, U. of Texas, Austin, TX.
- Belady, L. A. (1966). A study of replacement algorithms for a virtual-storage computer. *IBM Systems Journal*, 5(2).
- Belady, L. A., Nelson, R. A., & Shedler, G. S. 1969. An anomaly in space-time characteristics of certain programs running in a paging machine. *Communications of the ACM*, 12(6), 349 - 353.
- Deitel, P., & Choffnes, D. (2004). *Operating systems*. Prentice Hall.
- Dhamdhere, D. (2008). *Operating systems*. McGraw-Hill.
- Franaszek, P. A., & Wagner, T. J. (1974). Some distribution-free aspects of paging algorithm performance. *Journal of the ACM*, 21(1).
- Fine, G. H., Jackson, C. W., & McIsaac, P. V. (1966). Dynamic program behavior under paging. *Proceedings of the 1966 21st National ACM Annual Conference*, pp. 223 – 228.
- Johnson, N. L., Kotz, S., & Balakrishnan, N. (2005). *Univariate Discrete Distributions* (3rd ed). Wiley-Interscience.
- Keromytis, A. D., Wright, J. L., De Raadt, T. D., & Burnside, M. (2006). Cryptography as an operating system service: A case study. *ACM Transactions on Computer Systems*, 24(1).
- Lions, J. (1978). An operating systems case study. *ACM SIGOPS Operating Systems Review*, 12(3).
- Mattson, R. L., Gecsei, J., Sultz, D. R., & Traiger, I. L. (1970). Evaluation techniques for storage hierarchies. *IBM Systems Journal*, 9(2).
- McMaster, K., Sambasivam, S., & Anderson, N. (2009a). How anomalous is Belady's anomaly? *Issues in Informing Science and Information Technology*, 6, 825-836. Retrieved from <http://iisit.org/Vol6/IISITv6p825-836McMaster617.pdf>
- McMaster, K., Sambasivam, S., & Anderson, N. 2009b. A probability model for Belady's anomaly. In *The Proceedings of the Information Systems Education Conference 2009*, Vol 26 (Washington DC).
- Randell, B., & Kuehner, C. J. (1968). Dynamic storage allocation systems, *Communications of the ACM*, 11(5).
- Schlesinger, R., & Garrido, J. (2007). *Principles of modern operating systems*. Jones & Bartlett.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2008). *Operating system concepts* (8th ed.). Wiley.
- Snedecor, G. W., & Cochran, W. G. (1989). *Statistical methods* (8th ed.). Iowa State University Press.
- Stuart, B. (2008). *Principles of operating systems: Design and applications*. Course Technology.
- Warford, J. S. (2009). *Computer systems* (4th ed.). Jones & Bartlett Publishers.

## Biographies



Dr. Kirby McMaster recently retired from the CS Dept. at Weber State University. His primary research interests are in Database Systems, Discrete Mathematics, and Mathematical Frameworks for Computer Science.



**Dr. Samuel Sambasivam** is chairman of the CS Dept at Azusa Pacific University. His research interests include Optimization Methods, Expert Systems, Client/Server Applications, Database Systems, and Genetic Algorithms.



**Dr. Nicole Anderson** is an Assistant Professor in the CS Dept. at Winona State University. Her teaching and research interests involve Database Systems, Software Engineering, and the development of collaborative online learning communities.