The Segmentation Method for Long Line Optimization

Chuan Shi and Stanley B. Gershwin

Department of Mechanical Engineering
Massachusetts Institute of Technology

4th International Conference on Information Systems, Logistics and Supply Chain
Quebec, Canada

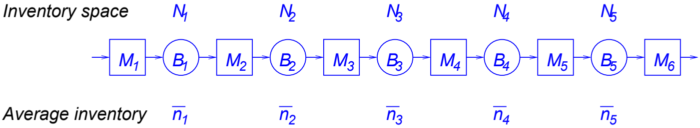
August 26-29, 2012

Content

1. Introduction
   - Background
   - Problem Statement
   - Motivation
2. The Segmentation Method for Long Line Optimization
   - Outline of the method
   - Balanced line example
   - The segmentation method
   - Intuitive explanation
   - Unbalanced line example
   - Reduction of the edge effect
   - Numerical experiments
3. Summary

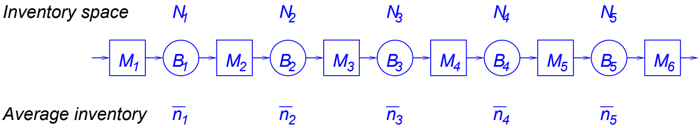
Background

A production line (or transfer line, flow line, etc.) is organized with unreliable machines connected in series and separated by buffers.



Background

A production line (or transfer line, flow line, etc.) is organized with unreliable machines connected in series and separated by buffers.



Benefits and costs of buffers:

- benefits: buffer decouple machines and mitigate the effect of a failure of one of the machines on the operation of others.
- costs: buffers create buffer space costs and inventory costs as well as longer lead times of products.

Optimizing the buffer space allocation makes factories most profitable.

Model Details

- All machines have unit operation times.
- Machine failure and repair times are geometrically distributed.

Background

Model Details

- All machines have unit operation times.
- Machine failure and repair times are geometrically distributed.
- $p_i = 1/\text{MTTF}_i$ = probability of failure of M_i during one operation time while it is up and not idle.
- $r_i = 1/\text{MTTR}_i$ = probability of repair of M_i during one operation time while it is down.

Background

Model Details

- All machines have unit operation times.
- Machine failure and repair times are geometrically distributed.
- $p_i = 1/\text{MTTF}_i$ = probability of failure of M_i during one operation time while it is up and not idle.
- $r_i = 1/\text{MTTR}_i$ = probability of repair of M_i during one operation time while it is down.
- Machines are forced to be idle when their upstream buffers are empty (starvation) and when their downstream buffers are full (blockage).
- The first machine is never starved and the last is never blocked.

Background

Analysis of lines

- Lines are analyzed by modeling them as Markov chains and finding their steady-state distributions.
- Production rates and average buffer levels are obtained from the steady-state distributions.

Background

Analysis of lines

- Lines are analyzed by modeling them as Markov chains and finding their steady-state distributions.
- Production rates and average buffer levels are obtained from the steady-state distributions.
- Two-machine lines have analytical solutions.

Background

Analysis of lines

- Lines are analyzed by modeling them as Markov chains and finding their steady-state distributions.
- Production rates and average buffer levels are obtained from the steady-state distributions.
- Two-machine lines have analytical solutions.
- Short lines can be analyzed numerically.

Background

Analysis of lines

- Lines are analyzed by modeling them as Markov chains and finding their steady-state distributions.
- Production rates and average buffer levels are obtained from the steady-state distributions.
- Two-machine lines have analytical solutions.
- Short lines can be analyzed numerically.
- Long lines require simulation or analytical approximation.
  - Simulation adds difficulties in optimization because of the random nature of the numerical results.
  - We use analytical approximation.

Problem Statement

Shi and Gershwin (2009) developed an efficient buffer design algorithm for production line profit maximization subject to a production rate constraint:

$$\max_{\mathbf{N}} \quad J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N}) \tag{1}$$

$$\text{s.t.} \quad P(\mathbf{N}) \geq \hat{P}$$

where $J(\mathbf{N})$ is the profit of the line, A (\$/part) is the revenue coefficient, while b_i and c_i (\$/part/time unit) are cost coefficients. $\hat{P}$ is the target production rate. That algorithm works for the production line models of Buzacott (1967), Tolio et al. (2002), and Levantesi et al. (2003).

Problem Statement

To solve (1), Shi and Gershwin (2009) start with solving a corresponding unconstrained problem

$$\max_{\mathbf{N}} \quad J(\mathbf{N}) \quad = \quad AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N}) \qquad (2)$$

Let $\mathbf{N}^u$ be the solution of the unconstrained problem (2). Then

- If $P(\mathbf{N}^u) \geq \hat{P}$, then the optimal solution of the original problem (1) $\mathbf{N}^\star = \mathbf{N}^u$;

Problem Statement

To solve (1), Shi and Gershwin (2009) start with solving a corresponding unconstrained problem

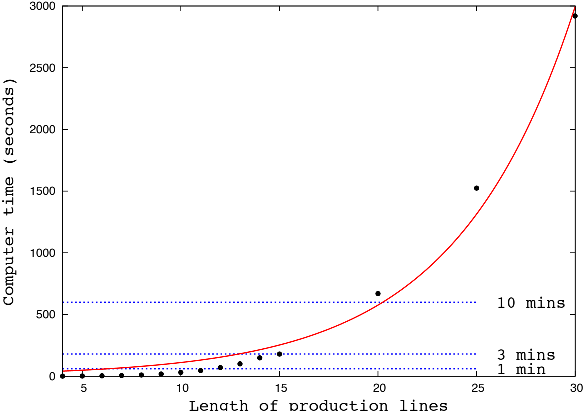
$$\max_{\mathbf{N}} \quad J(\mathbf{N}) \quad = \quad AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N}) \qquad (2)$$

Let $\mathbf{N}^u$ be the solution of the unconstrained problem (2). Then

- If $P(\mathbf{N}^u) \geq \hat{P}$, then the optimal solution of the original problem (1) $\mathbf{N}^\star = \mathbf{N}^u$;
- If $P(\mathbf{N}^u) < \hat{P}$, the algorithm conducts an one dimensional search in the revenue coefficient A and solves (2) iteratively for different As until it finds a value of A for which the solution of (2), say $\mathbf{N}^u(A)$, satisfies $P(\mathbf{N}^u(A)) = \hat{P}$. Then the optimal solution of (1) is $\mathbf{N}^\star = \mathbf{N}^u(A)$.

Motivation

The computer time of the algorithm increases exponentially with the length of the line when the one dimensional search in A is adopted to find $\mathbf{N}^\star = \mathbf{N}^u(A)$ such that $P(\mathbf{N}^u(A)) = \hat{P}$.



Motivation

Two factors in the algorithm that determine the computer time:

- the time required for solving (2) for some A in each iteration;
- the number of iterations.

As the length of the line increases, both factors increase and they lead to a drastic increase in computer time.

Therefore, it is desirable to find a method that reduces the computer time in long line optimization.

Content

1. Introduction
   - Background
   - Problem Statement
   - Motivation
2. The Segmentation Method for Long Line Optimization
   - Outline of the method
   - Balanced line example
   - The segmentation method
   - Intuitive explanation
   - Unbalanced line example
   - Reduction of the edge effect
   - Numerical experiments
3. Summary

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),
- optimizes these short lines separately,

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),
- optimizes these short lines separately,
- and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line.

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),
- optimizes these short lines separately,
- and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line.

The essence of the method is how the segments are chosen and how they are combined.

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),
- optimizes these short lines separately,
- and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line.

The essence of the method is how the segments are chosen and how they are combined.

The method greatly reduces computer time.

Outline of the Method

We propose a segmentation method. The method first solves the unconstrained problem. If $P(\mathbf{N}^u) < \hat{P}$, instead of optimizing the original long line, the method

- divides it into several short lines (segments),
- optimizes these short lines separately,
- and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line.

The essence of the method is how the segments are chosen and how they are combined.

The method greatly reduces computer time.

We do not have mathematical proof, rigorous bounds, etc. We only have numerical evidence of accuracy, and an intuitive explanation of why it works.

Balanced Line Example

Example: 20-machine line

- $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 20$;
- $b_i = c_i = 1, i = 1, \cdots, 19$;
- the revenue coefficient $A = 10,000$;
- $\hat{P} = .88$.

$$\max_{\mathbf{N}} J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N})$$

$$\text{s.t. } P(\mathbf{N}) \geq \hat{P}$$

Balanced Line Example

Example: 20-machine line

- $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 20$;
- $b_i = c_i = 1, i = 1, \cdots, 19$;
- the revenue coefficient $A = 10,000$;
- $\hat{P} = .88$.

$$\max_{\mathbf{N}} J(\mathbf{N}) = AP(\mathbf{N}) - \sum_{i=1}^{K-1} b_i N_i - \sum_{i=1}^{K-1} c_i \bar{n}_i(\mathbf{N})$$

$$\text{s.t. } P(\mathbf{N}) \geq \hat{P}$$

Three segmented 10-machine 9-buffer lines:

- $M_1 - B_1 - M_2 - \cdots - B_9 - M_{10}$;
- $M_6 - B_6 - M_7 - \cdots - B_{14} - M_{15}$;
- $M_{11} - B_{11} - M_{12} - \cdots - B_{19} - M_{20}$.

For these three 10-machine lines, we modify the revenue coefficient A to 5,000. The value of A is modified such that the production rate constraint will be satisfied with equality in all segmented lines as well.

Balanced Line Example

Let $\mathbf{N}_1^\star$, $\mathbf{N}_2^\star$, and $\mathbf{N}_3^\star$ be the optimal buffer distributions of the three lines:

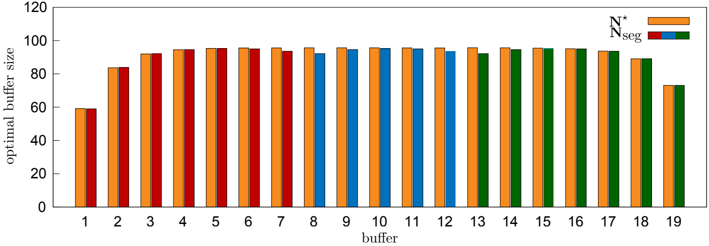
	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
$\mathbf{N}_1^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12
	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
$\mathbf{N}_2^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12
	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
$\mathbf{N}_3^\star$	59.00	83.89	92.16	94.63	95.20	94.97	93.63	89.15	73.12

We take some results from each segment and construct the approximate solution $\mathbf{N}_{\text{seg}}$ for the original line who has 19 buffers as follows:

$$\mathbf{N}_{\text{seg}} = \Big(\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_7), \mathbf{N}_2^\star(B_8), \mathbf{N}_2^\star(B_9), \cdots, \mathbf{N}_2^\star(B_{12}), \mathbf{N}_3^\star(B_{13}), \mathbf{N}_3^\star(B_{14}), \cdots, \mathbf{N}_3^\star(B_{19})\Big)$$

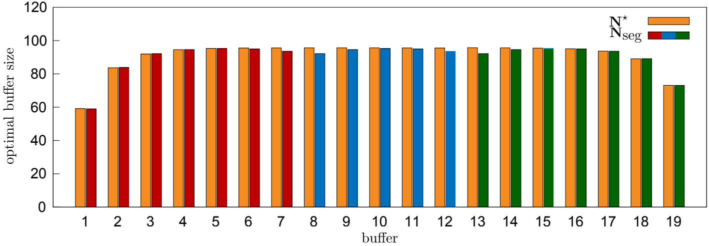
Balanced Line Example

For comparison, we optimize the original 20-machine 19-buffer line directly and let $\mathbf{N}^\star$ be the optimal buffer distribution.



Balanced Line Example

For comparison, we optimize the original 20-machine 19-buffer line directly and let $\mathbf{N}^\star$ be the optimal buffer distribution.



- computer time: 483.74 seconds vs. 125.00 seconds;
- production rate: $P(\mathbf{N}^\star) = .8800$, while $P(\mathbf{N}_{\text{seg}}) = .8798$;
- profit $J(\mathbf{N}^\star) = \$6259.11/\text{time unit}$, while $J(\mathbf{N}_{\text{seg}}) = \$6270.34/\text{time unit}$.

The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.

The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.
1. Choose the length of the segmented lines, say k.

The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.
1. Choose the length of the segmented lines, say k.
2. Choose the number of segments, say s.

The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.
1. Choose the length of the segmented lines, say k.
2. Choose the number of segments, say s.
3. Adjust the revenue coefficient A for the segmented lines. Optimize each segmented line and let $\mathbf{N}_i^\star$ be the optimal buffer distribution of the ith segment, $i = 1, 2, \cdots, s$.

The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.
1. Choose the length of the segmented lines, say k.
2. Choose the number of segments, say s.
3. Adjust the revenue coefficient A for the segmented lines. Optimize each segmented line and let $\mathbf{N}_i^\star$ be the optimal buffer distribution of the ith segment, $i = 1, 2, \cdots, s$.
4. **Construct $\mathbf{N_{seg}}$ from $\mathbf{N}_i^\star$s. If a buffer is contained in more than one segment, the largest value of that buffer size from any segment is used in $\mathbf{N_{seg}}$.**

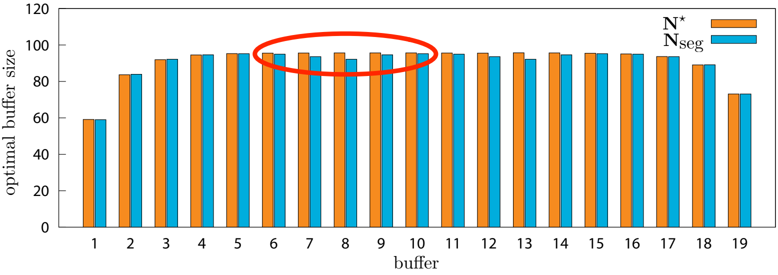
The Segmentation Method

To solve (1) for a K-machine $K-1$-buffer line, the segmentation method consists of the following steps:

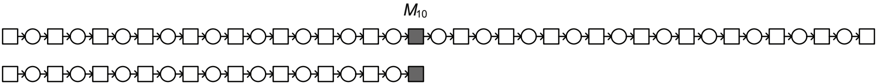
0. Solve the unconstrained problem (2) . If the solution satisfies the production rate constraint, stop. Otherwise, go to step 1.
1. Choose the length of the segmented lines, say k.
2. Choose the number of segments, say s.
3. Adjust the revenue coefficient A for the segmented lines. Optimize each segmented line and let $\mathbf{N}_i^\star$ be the optimal buffer distribution of the ith segment, $i = 1, 2, \cdots, s$.
4. **Construct $\mathbf{N}_{\mathbf{seg}}$ from $\mathbf{N}_i^\star$s. If a buffer is contained in more than one segment, the largest value of that buffer size from any segment is used in $\mathbf{N}_{\mathbf{seg}}$.**
5. Verify that $P(\mathbf{N}_{\text{seg}}) \approx \hat{P}$ and compute the profit of the line by $J(\mathbf{N}_{\text{seg}})$.

The Segmentation Method

- The buffer distribution derived by segmentation approximates the actual optimal buffer distribution of the original very well.
- We observe small discrepancy between $\mathbf{N}_{\text{seg}}$ and $\mathbf{N}^{\star}$, especially in buffers shared by two adjacent segments.

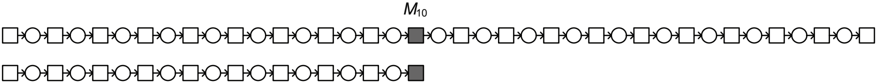


Intuitive Explanation



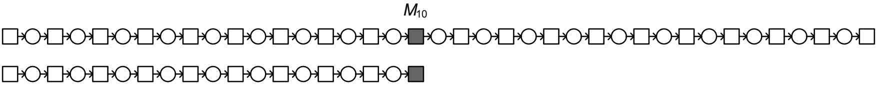
- There are 10 machines and 10 buffers downstream of M_{10} in the original line, while there is nothing downstream of M_{10} in the segmented line.

Intuitive Explanation



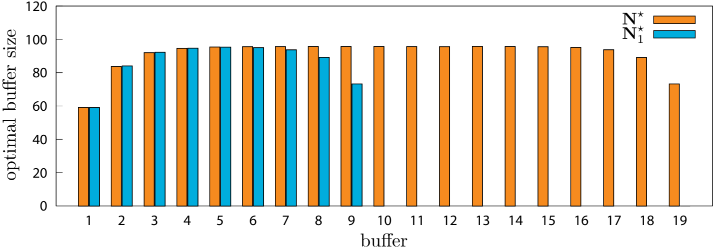
- There are 10 machines and 10 buffers downstream of M_{10} in the original line, while there is nothing downstream of M_{10} in the segmented line.
- In the segmented line, M_{10} will not be blocked, while in the original line M_{10} can be blocked if buffer B_{10} becomes full because of downstream machine failures.

Intuitive Explanation



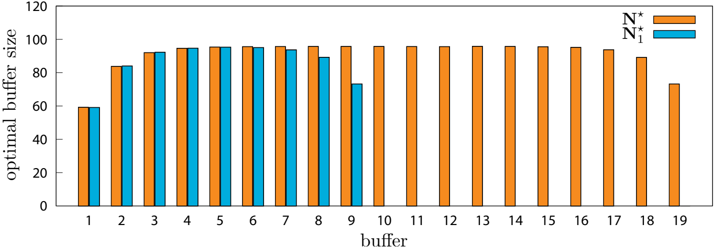
- There are 10 machines and 10 buffers downstream of M_{10} in the original line, while there is nothing downstream of M_{10} in the segmented line.
- In the segmented line, M_{10} will not be blocked, while in the original line M_{10} can be blocked if buffer B_{10} becomes full because of downstream machine failures.
- The variability downstream of M_{10} in the original line is not 0, while the variability downstream of M_{10} in the segmented line is 0.

Intuitive Explanation



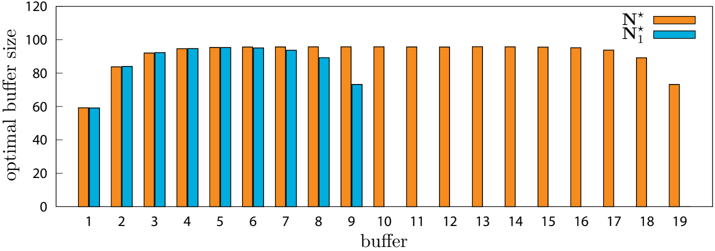
- B_9, B_8, B_7, and B_6 in the segmented line are reduced because of the zero variability downstream of M_{10}.

Intuitive Explanation



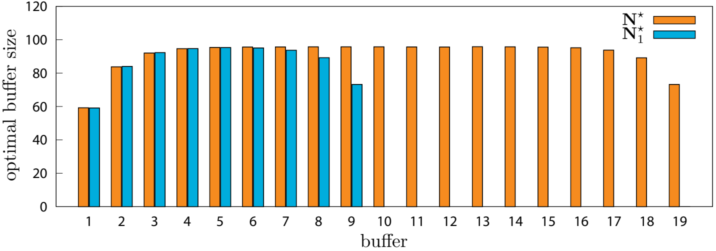
- B_9, B_8, B_7, and B_6 in the segmented line are reduced because of the zero variability downstream of M_{10}.
- No visible difference in buffers B_1 to B_5 because the benefits (in terms of reduced buffer space) brought by zero variability downstream of M_{10} in the segmented line have been consumed by buffers B_6 to B_9.

Intuitive Explanation



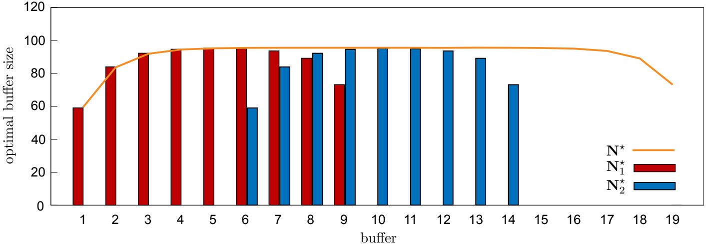
- B_1 to B_5 should be kept large enough to achieve the target production rate. This emphasizes the importance of the condition that the production rate constraint has to be active.

Intuitive Explanation



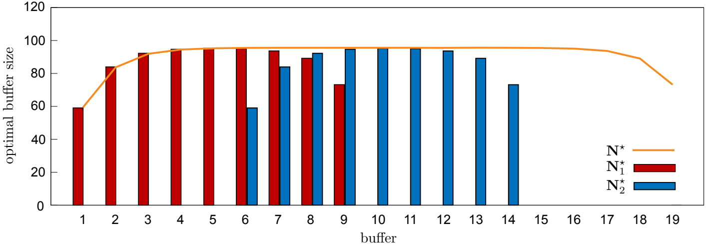
- B_1 to B_5 should be kept large enough to achieve the target production rate. This emphasizes the importance of the condition that the production rate constraint has to be active.
- The analysis reveals the source of inaccuracy of $\mathbf{N}_{seg}$ as compared to the $\mathbf{N}^\star$. To explain this, we refer to the benefits brought by the zero variability to buffers at the end(s) of a segment as the *edge effect*.

Intuitive Explanation



- For B_6 to B_9, we choose $\max\{\mathbf{N}_1^\star(B_i), \mathbf{N}_2^\star(B_i)\}, i = 6, 7, 8, 9$ as their sizes in $\mathbf{N}_{\text{seg}}$.

Intuitive Explanation



- For B_6 to B_9, we choose $\max\{\mathbf{N}_1^\star(B_i), \mathbf{N}_2^\star(B_i)\}$, $i = 6, 7, 8, 9$ as their sizes in $\mathbf{N}_{\text{seg}}$.
- This setting mitigates the edge effect the most and gives the best approximation of the first nine components of $\mathbf{N}^\star$.

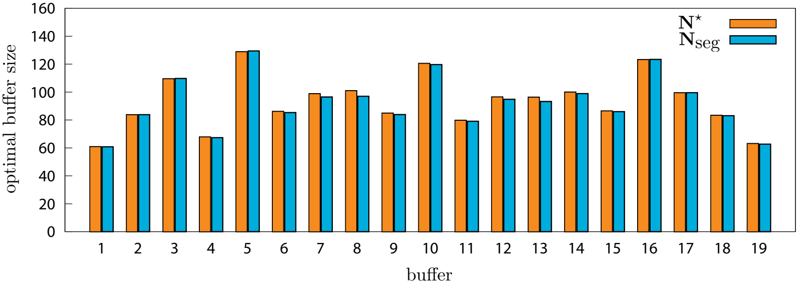
Unbalanced Line Example

- $r_i = .1$ and $p_i = .01$, $i = 1, \cdots, 20$;
- buffer coefficients b_i and c_i are listed in the table below;
- the revenue coefficient $A = 10,000$;
- $\hat{P} = .88$.

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
0.26	1.28	0.56	1.92	0.32	1.92	1.62	0.86	1.60	1.32
b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	
1.70	1.36	1.50	1.32	1.42	0.56	0.20	1.40	1.92	

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
1.84	0.20	1.10	1.94	1.96	0.98	0.30	1.84	1.92	0.08
c_{11}	c_{12}	b_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	
1.88	1.52	0.80	0.36	0.08	0.10	1.66	0.64	0.08	

Unbalanced Line Example



- computer time: 603.33 seconds vs. 199.80 seconds;
- production rate: $P(\mathbf{N}^\star) = .8800$, while $P(\mathbf{N}_{\text{seg}}) = .8798$;
- profit $J(\mathbf{N}^\star) = \$5904.10/\text{time unit}$, while $J(\mathbf{N}_{\text{seg}}) = \$5922.54/\text{time unit}$.

Reduction of the Edge Effect

The edge effect can be further mitigated by increasing the number of buffers shared by two adjacent segments. This can be achieved by increasing

- the number of segments;
- or the length of each segment.

Reduction of the Edge Effect

The edge effect can be further mitigated by increasing the number of buffers shared by two adjacent segments. This can be achieved by increasing

- the number of segments;
- or the length of each segment.

For the following discussion, define $\mathbf{N}_{\text{seg}}(s, k)$ as the approximate buffer distribution for the original line resulted from the segmentation method with s k-machine line segments. In addition, let

$$e_{\text{seg}}(s, k) = \mathbf{N}_{\text{seg}}(s, k) - \mathbf{N}^{\star}$$

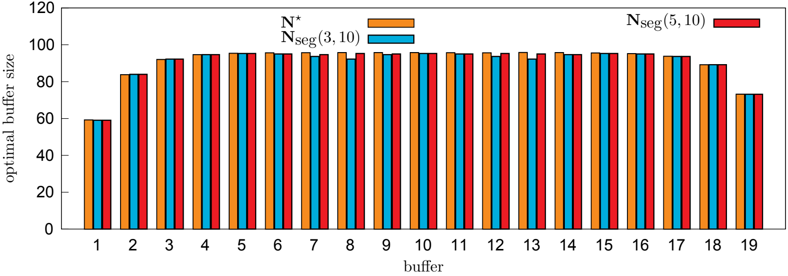
Reduction of the Edge Effect

- effect of the number of segments:
  - $M_1 - B_1 - M_2 - \cdots - B_9 - M_{10}$;
  - $M_4 - B_4 - M_5 - \cdots - B_{12} - M_{13}$;
  - $M_6 - B_6 - M_7 - \cdots - B_{14} - M_{15}$;
  - $M_8 - B_8 - M_9 - \cdots - B_{16} - M_{17}$;
  - $M_{11} - B_{11} - M_{12} - \cdots - B_{19} - M_{20}$.

$$\mathbf{N}_{\text{seg}} = \Big(\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_6), \mathbf{N}_2^\star(B_7), \mathbf{N}_2^\star(B_8), \mathbf{N}_2^\star(B_9),$$
$$\mathbf{N}_3^\star(B_{10}), \mathbf{N}_3^\star(B_{11}), \mathbf{N}_4^\star(B_{12}), \mathbf{N}_4^\star(B_{13}), \mathbf{N}_5^\star(B_{14}), \cdots, \mathbf{N}_5^\star(B_{19})\Big)$$

Reduction of the Edge Effect

- effect of the number of segments:



Reduction of the Edge Effect

- effect of the number of segments:

	$P(\mathbf{N})$	$J(\mathbf{N})$	computer time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	6259.11	483.74	—
$\mathbf{N}_{\text{seg}}(3,10)$	.8798	6270.34	125.00	-3.56
$\mathbf{N}_{\text{seg}}(5,10)$	.8799	6263.13	164.86	-1.02

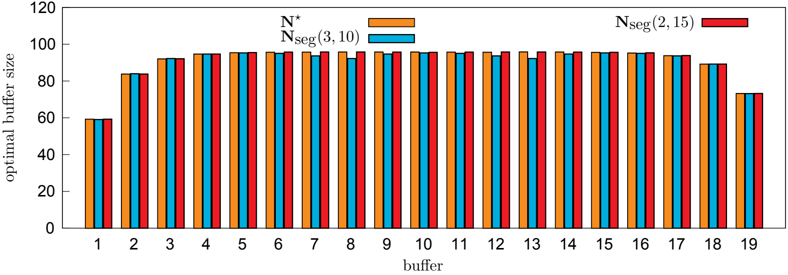
Reduction of the Edge Effect

- effect of the length of segments:
  - $M_1 - B_1 - M_2 - \cdots - B_{14} - M_{15}$.
  - $M_6 - B_6 - M_7 - \cdots - B_{19} - M_{20}$.

$$\mathbf{N}_{\text{seg}} = \Big(\mathbf{N}_1^\star(B_1), \mathbf{N}_1^\star(B_2), \cdots, \mathbf{N}_1^\star(B_{10}), \mathbf{N}_2^\star(B_{11}), \mathbf{N}_2^\star(B_{12}), \cdots, \mathbf{N}_2^\star(B_{19}) \Big)$$

Reduction of the Edge Effect

- effect of the length of segments:



Reduction of the Edge Effect

- effect of the length of segments:

	$P(\mathbf{N})$	$J(\mathbf{N})$	computer time (sec.)	max. buffer difference
$\mathbf{N}^\star$	.8800	6259.11	483.74	—
$\mathbf{N}_{\text{seg}}(3, 10)$	.8798	6270.34	125.00	−3.56
$\mathbf{N}_{\text{seg}}(2, 15)$	.8800	6257.03	340.71	0.19

Reduction of the Edge Effect

- To reduce the edge effect, increase the number of buffers shared by adjacent segments.

Reduction of the Edge Effect

- To reduce the edge effect, increase the number of buffers shared by adjacent segments.
- Increasing the number of segments or the length of segments improves the accuracy of the segmentation method at the cost of increasing computer time.

Reduction of the Edge Effect

- To reduce the edge effect, increase the number of buffers shared by adjacent segments.
- Increasing the number of segments or the length of segments improves the accuracy of the segmentation method at the cost of increasing computer time.
- The experiments studied here indicate that 10 or 15 appears to be a good choice for the length of a segment.

Reduction of the Edge Effect

- To reduce the edge effect, increase the number of buffers shared by adjacent segments.
- Increasing the number of segments or the length of segments improves the accuracy of the segmentation method at the cost of increasing computer time.
- The experiments studied here indicate that 10 or 15 appears to be a good choice for the length of a segment.
- The number of segments should be chosen such that every two adjacent segments should share a set of buffers. The number of buffers shared should be large enough (e.g. 4 buffers here) to eliminate the edge effect.

Numerical Experiments

- Study one hundred 20-machine 19-buffer lines that are randomly generated according to the method of Gershwin (2011).
- The isolated efficiency of every machine is greater than .89. $\hat{P} = .87$, and $P(\mathbf{N}^\star) \geq \hat{P}$ proves to be active in all cases.
- We segment each 20-machine line into three 10-machine lines and study three types of error:

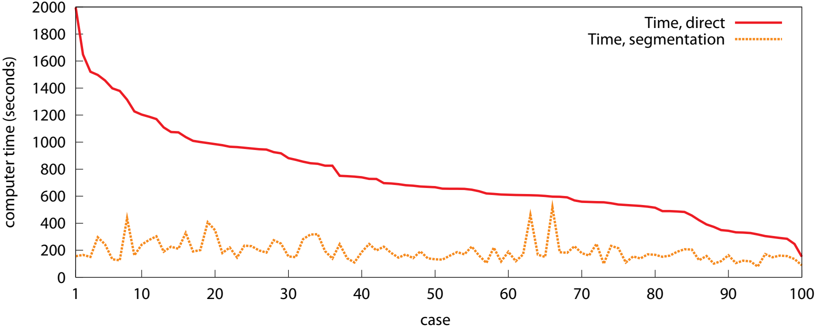
$$P_{\text{err}} = \left| \frac{P(\mathbf{N}^\star) - P(\mathbf{N}_{\text{seg}})}{P(\mathbf{N}^\star)} \right| \times 100\%$$

$$J_{\text{err}} = \left| \frac{J(\mathbf{N}^\star) - J(\mathbf{N}_{\text{seg}})}{J(\mathbf{N}^\star)} \right| \times 100\%$$

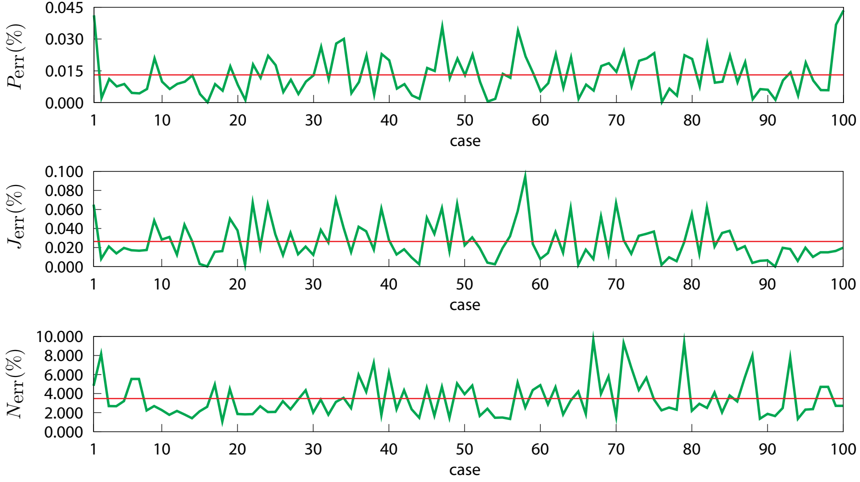
$$N_{\text{err}} = \max_{i=1,\cdots,19} \left\{ \left| \frac{\mathbf{N}^\star(B_i) - \mathbf{N}_{\text{seg}}(B_i)}{\mathbf{N}^\star(B_i)} \right| \times 100\% \right\}$$

Numerical Experiments

The average computer time for optimizing these 100 lines by the original method is 696.07 seconds, while the average computer time by segmentation is only 286.69 seconds.



Numerical Experiments



Numerical Experiments

$r_i = .1$ and $p_i = .01$. $b_i = c_i = 1$. The revenue coefficients A are $25,000$ and $30,000$ and $\hat{P} = .88$ for both cases. 10-machine line segments are used in segmentation.

The 50-machine 49-buffer line				
	$P(\mathbf{N})$	Profit ($)	computer time (sec.)	max. buffer error %
$\mathbf{N}^\star$	.8800	15218.95	8967.80	—
$\mathbf{N}_{\text{seg}}$	.8799	15261.12	1997.40	2.43%

The 60-machine 59-buffer line				
	$P(\mathbf{N})$	Profit ($)	computer time (sec.)	max. buffer error %
$\mathbf{N}^\star$	.8800	18209.24	17908.25	—
$\mathbf{N}_{\text{seg}}$	.8799	18260.85	2258.09	1.63%

Summary

We have developed a segmentation method for reducing computation time in long line optimization. Instead of optimizing the original long line, the method

- divides it into several short lines,
- optimizes these short lines separately,
- and combines the optimal buffer distributions to find an approximately optimal buffer distribution of the original line.

Numerical experience shows that the method saves substantial computer time and is accurate.

Future Research

- Rigorous bounds on accuracy
- Systematic guidelines for number and length of segments
- Extension to tree-structured assembly/disassembly systems
- Extension to single- and multiple-loop systems
- Machine selection