

The optimal implementation of n FIFO-queues
in single-level memory.

Prof. A.V. Sokolov, A.V. Drac
*IAMR KarRC RAS, Petrozavodsk State University,
Petrozavodsk.*

Introduction

In many applications there is a problem of allocation of multiple queues in single-level memory. There are two fundamentally different ways of organizing work with dynamic data structures – consecutive and linked allocation.

In the case of sequential presentation all the memory is splitted into several parts and each queue is allocated in separate section of memory. In this case we will have losses of memory when any queue overflow and other data structures don't overflow.

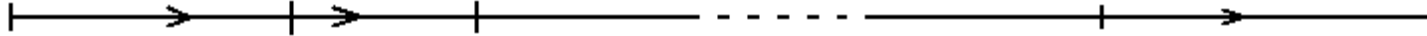


Figure 1: The consecutive implementation

The linked list implementation is the second method. In this view the data structure is stored as a list. In this case any number of elements of the lists can coexist in the fast memory area until the free memory is exhausted. But on the other hand, this approach requires an additional link field for each element.

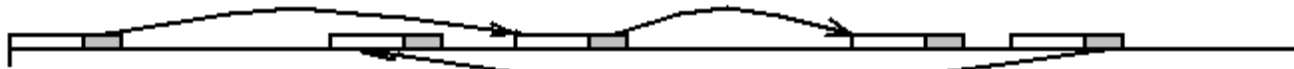


Figure 2: The linked list implementation

The problem

We suppose that if any queue allocates all its memory then all elements will be discarded until the appearance of the free memory (i.e. until the exclusion from queue). Overflow is a typical not an emergency situation. Such behaviour is called the “reset tail“.

We also suppose that the time is discrete and in the case of exclusion from an empty data structure there is no shutdown.

We will adhere to the following notation:

- m – size of memory,
- n – number of queues in the memory,
- p_j – probability of insertion of an element into the j -th queue,
- q_j – probability of deletion of an element from the j -th queue,
- r – probability of operation which doesn't change the data structure length (for example only reading),
- k_j – size of memory separated to the queue with number j with sequential presentation,
- l – the ratio of the size of node to the size of a pointer (for the linked presentation),
- P^* – the part of time which the system is situated in the state of “reset tail“.

We assume and the probabilities p_j , q_j , r are known. The problem is to find the part of time which the system is situated in the state of “reset tail“ and find the optimal partition of memory in case of consecutive presentation.

One queue

Consider one FIFO-queue in the memory of size m and build a homogeneous Markov's chain with $m + 2$ states, where states with numbers $0 \dots m$ correspond to the number of elements in queue and the state with number $m + 1$ corresponds to the state of "reset tail". Build a matrix of transition probabilities P :

$$P = \begin{pmatrix} q+r & p & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ q & r & p & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & q & r & p & \dots & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & q & r & p & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & q & r & p & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & q & r & p \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & q & r & p \end{pmatrix}$$

The component α_{m+1} of the limit vector is needed to be found from the equations ($\alpha P = \alpha$, $\alpha_0 + \dots + \alpha_{m+1} = 1$). The part of time which system is situated in the state of "reset tail" is equal to:

$$\alpha_{m+1} = \begin{cases} \frac{q-p}{\left(\frac{q}{p}\right)^{m+1} - 1} & q \neq p \\ \frac{1}{m+1} & q = p \end{cases} \quad (1)$$

Consecutive allocation of the queues

Consider k_1, \dots, k_n is the fixed partition of the memory.

As the mathematical model we consider the random walk on an integer lattice space inside n -dimensional parallelepiped with vertex at the origin, edges parallel to the axes and the lengths of edges k_1, \dots, k_n .

(i_1, \dots, i_n) is the state of the system. $0 \leq i_1 \leq k_1 + 1, \dots, 0 \leq i_n \leq k_n + 1$.

$i_j = k_j + 1$ – the states of “reset tail”.

$\alpha_{i_1 \dots i_n}$ it the limit probability which system is situated in the state (i_1, \dots, i_n) .

The problem is to find the part of time from the equations ($\alpha P = \alpha, \|\alpha\|_1 = 1$).

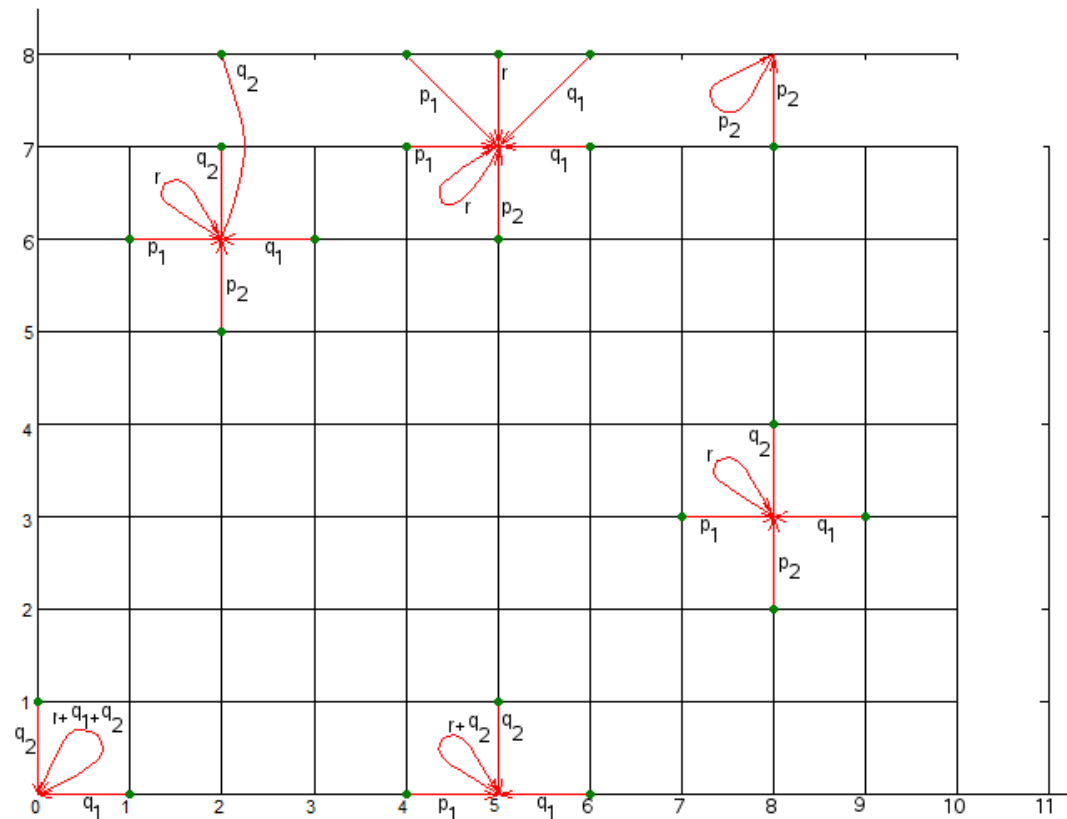


Figure 3: $n = 2, m = 17, k_1 = 10, k_2 = 7$

Consecutive allocation of the queues

$$I_1 = \{j : i_j = 0\}$$

$$I_2 = \{j : 1 \leq i_j \leq k_j - 2\}$$

$$I_3 = \{j : i_j = k_j - 1\}$$

$$I_4 = \{j : i_j = k_j\}$$

$$I_5 = \{j : i_j = k_j + 1\}$$

The solve of the equation is:

$$\alpha_{i_1 \dots i_n} = C \left(\frac{p_1}{q_1} \right)^{i_1} \dots \left(\frac{p_n}{q_n} \right)^{i_n} \left(1 - \sum_{j \in I_4} p_j \right) \quad I_5 = \emptyset$$

$$\alpha_{i_1 \dots i_j \dots i_n} = C \left(\frac{p_1}{q_1} \right)^{i_1} \dots \left(\frac{p_j}{q_j} \right)^{i_j - 1} \dots \left(\frac{p_n}{q_n} \right)^{i_n} p_j \quad I_5 = \{j\}$$

The constant C can be found from the normalization equation. The part of time which the system is situated in the state of “reset tail“ is equal to:

$$P^* = \sum_{i=1}^n p_i^* = \sum_{i=1}^s \frac{q_i - p_i}{\left(\frac{q_i}{p_i} \right)^{k_i + 1} - 1} + \sum_{i=s+1}^n \frac{p_i}{k_i + 1}$$

The optimal partition of memory

1. The case of equal probabilities $p_i = q_i$. The problem is to find

$$\min_{k_1 + \dots + k_n = m} \sum_{i=1}^n \frac{p_i}{k_i + 1}$$

Consider the function

$$F(k_1, k_2, \dots, k_{n-1}) = \sum_{i=1}^{n-1} \frac{p_i}{k_i + 1} + \frac{p_n}{m - \sum_{i=1}^{n-1} k_i + 1}$$

and find its minimum considering conditions $k_i > 0$, $k_1 + \dots + k_{n-1} < m$:

$$k_i^* = \frac{(m+n)\sqrt{p_i}}{\sum_{l=1}^n \sqrt{p_l}} - 1 \quad 1 \leq i \leq n-1$$

The minimum part of time which the system is situated in the state of “reset tail“ is equal to:

$$\sum_{i=1}^n \frac{p_i}{k_i^* + 1} = \sum_{i=1}^n \left(\frac{1}{m+n} \sqrt{p_i} \sum_{l=1}^n \sqrt{p_l} \right) = \frac{\left(\sum_{i=1}^n \sqrt{p_i} \right)^2}{m+n}$$

The optimal partition of memory

2. The general case.

Denote Z_m – part of time which system is situated in the state of “reset tail“ in the case of optimal partition of memory of size m .

$$Z_m(k_1, \dots, k_n) = \min_{k_1 + \dots + k_n = m} \sum_{i=1}^n p_i^*(k_i)$$

p_i^* is the part of time which i -th queue is situated in the state of “reset tail“.

Consider the recurrence relation

$$Z_m(k_1, \dots, k_n) = Z_{m-1} - \max_{1 \leq i \leq n} (p_i^*(k_i) - p_i^*(k_i + 1))$$

The start value Z_0 is equal to

$$Z_0(0, \dots, 0) = \sum_{i=1}^n p_i$$

I.e. in the case of lack of memory, any attempt of insertion will lead to the lost of element.

Using this scheme it is possible to calculate the optimal partition of memory and part of time which system is situated in the state of “reset tail“ over the time $O(mn)$

The linked list implementation

As the mathematical model we consider the random walk on the integer lattice in n -dimensional pyramid with edges $0 \leq i_1 \leq M, 0 \leq i_2 \leq M, \dots, 0 \leq i_n \leq M$ and base $i_1 + i_2 + \dots + i_n = M$.
 ($M = m(1 - \frac{1}{l})$)
 (i_1, \dots, i_n) is the state of the system.
 The problem is to find the part of time from the equations ($\alpha P = \alpha, \|\alpha\|_1 = 1$).

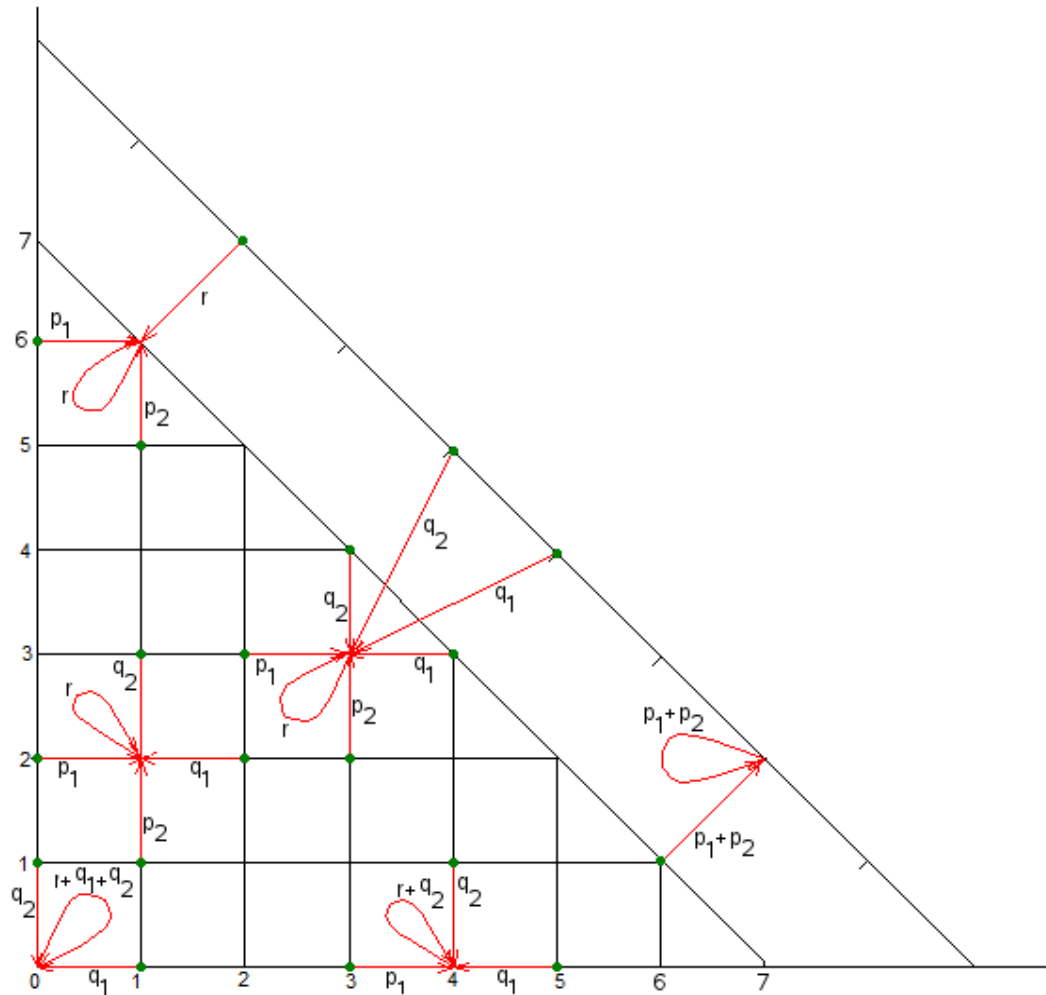


Figure 4: $n = 2, M = 7$

The linked list implementation

The solve of the equation is:

$$\alpha_{i_1 i_2 \dots i_n} = C \left(\frac{p_1}{q_1} \right)^{i_1} \dots \left(\frac{p_n}{q_n} \right)^{i_n} \quad 0 \leq i_1 + \dots + i_n \leq M - 1$$

$$\alpha_{i_1 i_2 \dots i_n} = C(1 - p_1 - \dots - p_n) \left(\frac{p_1}{q_1} \right)^{i_1} \dots \left(\frac{p_n}{q_n} \right)^{i_n} \quad i_1 + \dots + i_n = M$$

$$\overline{\alpha_{i_1 i_2 \dots i_n}} = C(p_1 + \dots + p_n) \left(\frac{p_1}{q_1} \right)^{i_1} \dots \left(\frac{p_n}{q_n} \right)^{i_n} \quad i_1 + \dots + i_n = M$$

The constant C can be found from the normalization equation.

1. The case of different probabilities: $p_i/q_i \neq p_j/q_j$ and $p_i \neq q_i$

Denote $x_i = p_i/q_i$, then the part of time which system is situated in the state of “reset tail“ is equal to:

$$P^* = (p_1 + \dots + p_n) \frac{\sum_{i=1}^n \frac{x_i^{M+n-1}}{\prod_{\substack{j=1 \\ j \neq i}}^n (x_i - x_j)}}{\sum_{i=1}^n \frac{x_i^{M+n}}{(x_i - 1) \prod_{\substack{j=1 \\ j \neq i}}^n (x_i - x_j)}} + \frac{1}{\prod_{j=1}^n (1 - x_j)}$$

The linked list implementation

2. The general case:

There is k_0 queues where $p_i = q_i$.

There is k_1 queues where $p_{i_1}/q_{i_1} = x_1$.

...

There is k_s queues where $p_{i_s}/q_{i_s} = x_s$.

The part of time which the system is situated in the state of “reset tail“ is equal to:

$$P^* = k_0(p_0k_0 + \dots + p_s k_s) \frac{\frac{\partial^{n-s-1}}{\partial^{k_0-1}x_0 \partial^{k_1-1}x_1 \dots \partial^{k_s-1}x_s} \left\{ \sum_{i=0}^s \frac{x_i^{M+n-1}}{\prod_{\substack{j=0 \\ j \neq i}}^s (x_i - x_j)} \right\}}{\frac{\partial^{n-s}}{\partial^{k_0}x_0 \partial^{k_1-1}x_1 \dots \partial^{k_s-1}x_s} \left\{ \sum_{i=0}^s \frac{x_i^{M+n}}{\prod_{\substack{j=0 \\ j \neq i}}^s (x_i - x_j)} \right\}}$$

Comparison between consecutive and linked presentations

1. $p_1 > q_1$ and $\frac{p_1}{q_1} > \frac{p_i}{q_i}$ for $i = 2, \dots, n$.

Consecutive implementation:

$$P^* \rightarrow \sum_{i=1}^n \max(p_i - q_i, 0)$$

Linked list implementation:

$$P_l^* \rightarrow (p_1 + \dots + p_n) \left(\frac{p_1 - q_1}{p_1} \right) = (p_1 + \dots + p_n) \left(1 - \frac{q_1}{p_1} \right)$$

$P_c^* < P_l^*$ even when the size of memory is rather small

2. $p_i = q_i$ for $i = 1, \dots, n$.

$$P_c^* \leq \frac{n}{m + n}$$

$$P_l^* = \frac{n}{M + n}$$

$$P_c^* < P_l^*$$

3. $p_i < q_i$ for $i = 1, \dots, n$. and $\frac{p_1}{q_1} > \frac{p_i}{q_i}$

$$P_c^* \approx \frac{1}{\exp\left(\frac{m-n}{n} - \log n\right)} = O\left(\exp\left(-\frac{m}{\sum_{i=1}^n \log\left(\frac{q_i}{p_i}\right)}\right)\right)$$

$$P_l^* = O\left(\left(\frac{q_1}{p_1}\right)^M\right) = O\left(\exp\left(-m\left(1 - \frac{1}{l}\right) \log \frac{q_1}{p_1}\right)\right)$$