APPENDIX C

Statistical Analysis

The statistical analysis of the results of the developed algorithms is given here. These are applied to a particular topology from the category of cycles. To analyze the correctness of the results, both the approximate and optimised minimal values are needed. Since the method to do this has exponential complexity, it is only feasible to analyse simple and relatively small topologies on a PC.

The topology investigated here, is a cycle with one sink and one source. Both its paths contain two buffers each, as depicted in Figure 1.

![Diagram of a 4-node cycle]

Figure 1: A 4-node cycle.

In order to obtain statistical results, several combinations of consumption and production numbers are generated. All numbers are varied between 1 and 9 and their values computed according to the developed equations.
For each combination, the total minimal length is once computed approximately as $LA$ and once as the optimised length $LO$. The error is computed as $LA - LO$. The distribution of the errors over all cases is stored.

1.1 Computing the total minimal length

To obtain the minimal lengths, a dedicated scheduling mechanism is created. Then the nodes are scheduled one at a time as much times as their repetition rate indicates, according to the following criteria: the nodes are fired such that the schedule shifts the tokens from the source to the sink as much as possible. The sink is executed whenever possible, while the source is executed only when strictly needed. After every execution the resulting state, which corresponds to the enumeration of the live tokens in each buffer, is evaluated. The smallest result thus obtained is then kept as the final minimal total buffer storage required for that graph.

1.2 Computing the total minimal lengths $LA$ and $LO$

For the topology of Figure 1, no simplification of the general proposed algorithm for a cycle is possible. Both paths need to be tested for expansion requirements. When expansion is needed, it is computed for each buffer of the path and the best result is selected. Two methods are applied for computing the approximated value. In the first method, the total minimal storage requirement $LA$ according to lemma 5.3 is computed. In the second method an optimized result $LO$ according to lemma 5.4 is computed.
Out of a total of 4,30,46,720 combinations, the approximation yielded the optimised values for 4,02,22,292 cases, which corresponds to 93.44%. This shows that the approximate method yields close to accurate results. Moreover, for consistent conditions, which require no buffer expansion, the $L(\text{app})$ and $L(\text{opt})$ values of minimal buffer lengths come exactly equal to $L_{\text{min}}$ in all the cases.

Linear scale and Logarithmic scale representation of the error distribution is shown in Figure 2 and Figure 3. The routine generating these results is attached at the end.
Figure 2: Difference between $L(\text{app})$ and $L(\text{opt})$
Figure 3: Difference in $L_{\text{app}}$ and $L_{\text{opt}}$ on logarithmic scale

Series 1:

<table>
<thead>
<tr>
<th>Difference</th>
<th>Log of number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.604</td>
<td>8</td>
</tr>
<tr>
<td>6.101</td>
<td>7.5</td>
</tr>
<tr>
<td>5.831</td>
<td>7</td>
</tr>
<tr>
<td>5.531</td>
<td>6.5</td>
</tr>
<tr>
<td>5.398</td>
<td>6</td>
</tr>
<tr>
<td>5.032</td>
<td>5.5</td>
</tr>
<tr>
<td>5.036</td>
<td>5.5</td>
</tr>
<tr>
<td>4.52</td>
<td>5</td>
</tr>
<tr>
<td>4.292</td>
<td>4.5</td>
</tr>
<tr>
<td>3.865</td>
<td>4</td>
</tr>
<tr>
<td>3.155</td>
<td>3.5</td>
</tr>
<tr>
<td>2.62</td>
<td>3.5</td>
</tr>
<tr>
<td>2.324</td>
<td>3</td>
</tr>
<tr>
<td>2.099</td>
<td>2.5</td>
</tr>
<tr>
<td>2.653</td>
<td>2.5</td>
</tr>
<tr>
<td>2.342</td>
<td>2</td>
</tr>
<tr>
<td>2.436</td>
<td>1.5</td>
</tr>
<tr>
<td>1.53</td>
<td>1</td>
</tr>
<tr>
<td>1.477</td>
<td>1</td>
</tr>
</tbody>
</table>

Log of number of cases
```c
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
#include<math.h>
#include<alloc.h>

int gcd(int p, int q)
{
    int r;
    r=p-(p/q*q);
    if(r==0)
        return q;
    else
        gcd(q,r);
}

void main()
{
    const m=1000;
    const o=3;
    int i, j, temp;
    FILE *yes;
    yes=fopen("sa.txt","w");
    float n1=0, n2=0, n3=0, n4=0, p1=0, p2=0, p3=0, p4=0;
    float t1=0, t2=0, t3=0, t4=0;
    float c1=0, c2=0, c3=0, c4=0, n=9;
    float count=0, y=0, z=0;
    float e1, e2, e3, e4, e1A, e2A, e3A, e4A;
    float S1, S2, s1, s2, d1, d2, d3, d4;
    float s1A, S1A, S2A, s2A, sx1, sx2, sx3, sx4;
    float P1, C1, D1, P2, C2, D2;
    float lw, *lx, *ly, *lz;
    lw=(float*)malloc(sizeof (lw)*m);
    lx=(float*)malloc(sizeof (lx)*m);
    ly=(float*)malloc(sizeof (ly)*m);
    lz=(float*)malloc(sizeof (lz)*m);
    111=(float*)malloc(sizeof (111)*o);
    122=(float*)malloc(sizeof (122)*o);
    133=(float*)malloc(sizeof (133)*o);
    144=(float*)malloc(sizeof (144)*o);
    11A=(float*)malloc(sizeof (11A)*o);
    12A=(float*)malloc(sizeof (12A)*o);
    13A=(float*)malloc(sizeof (13A)*o);
    14A=(float*)malloc(sizeof (14A)*o);

    for(i=0; i<m; i++)
    {
        lz[i]=0;
    }
}
```

ly[i]=0;
lw[i]=0;
1x[i]=0;
}

111[0]=count;
122[0]=count;
133[0]=count;
144[0]=count;
l1A[0]=count;
l2A[0]=count;
l3A[0]=count;
l4A[0]=count;
count++;

for(c4=l;c4<=n;c4++)
{
    for(c2=l;c2<=n;c2++)
    {
        for(p4=l;p4<=n;p4++)
        {
            for(p2=l;p2<=n;p2++)
            {
                for(c3=l;c3<=n;c3++)
                {
                    for(c1=1;cl<=n;cl++)
                    {
                        for(p3=l;p3<=n;p3++)
                        {
                            for(p1=l;p1<=n;p1++)
                            {
                                d1=gcd(p1,c1);
d2=gcd(p2,c2);
d3=gcd(p3,c3);
d4=gcd(p4,c4);
l11[count]=p1+c1-d1;
l22[count]=p2+c2-d2;
l33[count]=p3+c3-d3;
l44[count]=p4+c4-d4;
l1A[count]=pl+cl-dl;
l2A[count]=p2+c2-d2;
l3A[count]=p3+c3-d3;
l4A[count]=p4+c4-d4;
sx1=111[count]-p1+d1;
sx2=122[count]-p2+d2;
sx3=133[count]-p3+d3;
sx4=144[count]-p4+d4;
S1=sx1/p1+(sx2/p2)*(c1/p1);
s1=((c1-d1)/p1)+((c2-d2)/p2)*(c1/p1);
S2=sx3/p3+(sx4/p4)*(c3/p3);
\[ s_2 = \frac{(c_3 - d_3)}{p_3} + \frac{(c_4 - d_4)}{p_4} \times \frac{c_3}{p_3}; \]

\[ P_1 = p_1 \times p_2; \]

\[ P_2 = p_3 \times p_4; \]

\[ C_1 = c_1 \times c_2; \]

\[ C_2 = c_3 \times c_4; \]

\[ D_1 = \gcd(P_1, C_1); \]

\[ D_2 = \gcd(P_2, C_2); \]

\[ S_{1A} = \frac{D_1 \times \lceil \left( p_2 \times s_{x1} + c_1 \times s_{x2} \right) / D_1 \rceil}{P_1}; \]

\[ s_{1A} = \frac{D_1 \times \lfloor \left( p_2 \times (c_1 - d_1) + c_1 \times (c_2 - d_2) \right) / D_1 \rfloor}{P_1}; \]

\[ S_{2A} = \frac{D_2 \times \lceil \left( p_4 \times s_{x3} + c_3 \times s_{x4} \right) / D_2 \rceil}{P_2}; \]

\[ s_{2A} = \frac{D_2 \times \lfloor \left( p_4 \times (c_3 - d_3) + c_3 \times (c_4 - d_4) \right) / D_2 \rfloor}{P_2}; \]

\[ \text{if}(S_1 \geq s_2) \]
\[ \{ \]
\[ \]e_1 = 0;
\[ e_2 = 0; \]
\[ \} \text{else} \]
\[ \{ \]
\[ \]e_1 = d_1 \times \lceil \left( p_1 / d_1 \right) \times (s_2 - S_1) \rceil; \]
\[ e_2 = d_2 \times \lfloor \left( p_2 / d_2 \right) \times \left( p_1 / d_1 \right) \times (s_2 - S_1) \rfloor; \]
\[ \} \]

\[ \text{if}(e_1 > e_2) \]
\[ 122[count] = 122[count] + e_2; \]
\[ \text{else} \]
\[ \{ \]
\[ \]if(e_1 == e_2) \]
\[ \{ \]
\[ \]if(l_{111}[count] < l_{122}[count]) \]
\[ l_{111}[count] = l_{111}[count] + e_1; \]
\[ \text{else} \]
\[ l_{122}[count] = l_{122}[count] + e_2; \]
\[ \} \]
\[ \text{else} \]
\[ l_{111}[count] = l_{111}[count] + e_1; \]
\[ y++; \]
\[ \}

\[ \text{if}(S_2 \geq s_1) \]
\[ \{ \]
\[ \]e_3 = 0;
\[ e_4 = 0; \]
\[ \} \text{else} \]
\[ \{ \]
\[ \]e_3 = d_3 \times \lceil \left( p_3 / d_3 \right) \times (s_1 - S_2) \rceil; \]
\[ e_4 = d_4 \times \lfloor \left( p_4 / d_4 \right) \times \left( p_3 / d_3 \right) \times (s_1 - S_2) \rfloor; \]
\[ \} \]
\[ \text{if}(e_3 > e_4) \]
\[ l_{144}[count] = l_{144}[count] + e_4; \]
\[ \text{else} \]
\[ \{ \]
if(e3==e4) {
if(133[count]<144[count])
133[count]=133[count]+e3;
else 144[count]=144[count]+e4;
} else {
133[count]=133[count]+e3;
}
++z;

if(S1A>=s2A) {
elA=0;
e2A=0;
} else {
elA=d1*(1+floor((p1/d1)*(s2A-S1A)));
e2A=d2*(1+floor((p2/d2)*(p1/d1)*(s2A-S1A)));
} if(e1A>e2A)
else {
if(e1A==e2A) {
if(11A[count]<12A[count])
}
++y;
}
if(S2A>=s1A) {
e3A=0;
e4A=0;
} else {
e3A=d3*(1+floor((p3/d3)*(s1A-S2A)));
e4A=d4*(1+floor((p4/d4)*(p3/d3)*(s1A-S2A)));
} if(e3A>e4A)
else {
200
if(e3A==e4A)
{
if(13A[count]<14A[count])
}
else
{
}
}
z++;

temp=111[1]-11A[1];
lw[temp]++;
temp=122[1]-12A[1];
lx[temp]++;
temp=133[1]-13A[1];
lx[temp]++;
temp=144[1]-14A[1];
lz[temp]++;

if((111[0])<(111[1]))
111[0]=111[1];

if(122[count-1]<122[count])
122[count-1]=122[count];

if(133[count-1]<133[count])
133[count-1]=133[count];

if(144[count-1]<144[count])
144[count-1]=144[count];

if((11A[0])<(11A[1]))
11A[0]=11A[1];

if(12A[count-1]<12A[count])

if(13A[count-1]<13A[count])

if(14A[count-1]<14A[count])

}
fprintf(yes,"Out of a total of %.0f cases \n",g);
fprintf(yes,"Maximum L1=%.0f, L2=%.0f, L3=%.0f, L4=%.0f\n", l1[0], l2[0], l3[0], l4[0]);
fprintf(yes,"Maximum L1A=%.0f, L2A=%.0f, L3A=%.0f, L4A=%.0f\n", l1[0], l2[0], l3[0], l4[0]);

for(i=0;i<m;i++)
{
    fprintf(yes," %.0f,%.0f,%.0f,%.0f have L app & L exact difference of %d n",lw[i],lx[i],ly[i],lz[i],i);
}

fclose(yes);
free(lw);
free(lx);
free(ly);
free(lz);
free(l11);
free(l12);
free(l13);
free(l14);
free(l11A);
free(l12A);
free(l13A);
free(l14A);