The Search for Consecutive Ones Submatrices: Faster and More General

Michael Dom and Rolf Niedermeier

Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany

Algorithms and Complexity in Durham 2007
Consecutive Ones Property (C1P)

A 0/1-matrix has the C1P if its columns can be permuted such that in each row the 1's form a block.
Consecutive Ones Property (C1P)

Example for a matrix having the C1P:

```
1 2 3 4 5
1
1 1 1 1 1
1 1 1 1
1 1 1
```
Consecutive Ones Property (C1P)

Example for a matrix having the C1P:

\[
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5 \\
1 & 1 & & & 1 \\
1 & 1 & & & 1 \\
1 & 1 & 1 & & \\
\end{array}
\]

\[
\begin{array}{cccccc}
 & & & & & \\
 & & & & & \\
 & & & & & \\
2 & 5 & 1 & 3 & 4 & \\
\end{array}
\]

\[
\begin{array}{cccc}
1 & 1 & 1 & \\
1 & 1 & 1 & \\
1 & 1 & 1 & \\
\end{array}
\]
Consecutive Ones Property (C1P)

Examples for matrices **not** having the C1P:

<table>
<thead>
<tr>
<th>1 1 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 0</td>
</tr>
<tr>
<td>0 0 1 1 0</td>
</tr>
<tr>
<td>1 0 0 1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1 1 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 0 0</td>
</tr>
<tr>
<td>0 0 1 1 0 0</td>
</tr>
<tr>
<td>0 0 0 1 1 0</td>
</tr>
<tr>
<td>1 0 0 0 1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1 1 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0</td>
</tr>
<tr>
<td>0 1 0 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1 1 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 1 0 0 0</td>
</tr>
<tr>
<td>0 0 0 1 1 1 0</td>
</tr>
<tr>
<td>1 0 1 0 1 0 0</td>
</tr>
</tbody>
</table>
Consecutive Ones Property (C1P)

The Consecutive Ones Property...

- ...expresses “locality” of the input data.
- ...appears in many applications, e.g.
  - in railway system optimization
    [Ruf, Schöbel, Discrete Optimization, 2004; Mecke, Wagner, ESA '04],
  - bioinformatics
    [Christof, Oswald, Reinelt, IPCO '98; Lu, Hsu, J. Comp. Biology, 2003].

- ...can be recognized in polynomial time

- ...is subject of current research
Min-COS-C (Min-COS-R)

*Given:* A matrix $M$ and a positive integer $k$.

*Question:* Can we delete at most $k$ columns (at most $k$ rows) such that the resulting matrix has the C1P?
Known and New Results

Min-COS-C:

- NP-hard for $(2, 3)$- and $(3, 2)$-matrices
- Approximation algorithms for maximization version on $(2, 3)$-, $(3, 2)$-, and $(2, *)$-matrices
- FPT and approximation results for $(*, 2)$-, $(2, *)$- and $(*, \Delta)$-matrices

Min-COS-R:

- NP-hard for $(3, 2)$-matrices

---

1 [Tan, Zhang, ISAAC '04]
2 [Dom, Guo, Niedermeier, TAMC '07]
Known and New Results

Min-COS-C:

- NP-hard for (2, 3)- and (3, 2)-matrices
- Approximation algorithms for maximization version on (2, 3)-, (3, 2)-, and (2, ∗)-matrices
- FPT and approximation results for (∗, 2)-, (2, ∗)- and (∗, Δ)-matrices
- Improved results for (∗, Δ)-matrices (FPT w.r.t (k, Δ))

Min-COS-R:

- NP-hard for (3, 2)-matrices
- FPT and approximation results for (∗, Δ)-matrices

---

1 [Tan, Zhang, ISAAC '04]
2 [Dom, Guo, Niedermeier, TAMC '07]
Structure of What Follows

- Algorithmic Framework
- From Circ1P to C1P
**Theorem:** A matrix has the C1P iff it contains none of the shown matrices.

[Tucker, Journal of Combinatorial Theory (B), 1972]
Min-COS-C / Min-COS-R on \((\ast, \Delta)\)-Matrices

\[
\begin{array}{cccccc}
1 & 1 & 0 & \cdots & 0 \\
0 & 1 & 1 & 0 & \cdots & 0 \\
\vdots & & & & & \\
0 & \cdots & 0 & 1 & 1 \\
1 & 0 & \cdots & 0 & 1 \\
\end{array}
\]

\(M_{I_p}, p \geq 1\)

\[
\begin{array}{cccccc}
1 & 1 & 0 & \cdots & 0 & 0 \\
0 & 1 & 1 & 0 & \cdots & 0 \\
0 & \cdots & 0 & 1 & 1 \\
1 & \cdots & 1 & 0 & 1 \\
\end{array}
\]

\(M_{II_p}, p \geq 1\)

\[
\begin{array}{cccccc}
1 & 1 & 0 & \cdots & 0 & 0 \\
0 & 1 & 1 & 0 & \cdots & 0 \\
0 & \cdots & 0 & 1 & 1 \\
1 & \cdots & 1 & 0 & 1 \\
\end{array}
\]

\(M_{III_p}, p \geq 1\)

\[
\begin{array}{cccccc}
1 & 1 & 0 & \cdots & 0 & 0 \\
0 & 1 & 1 & 0 & \cdots & 0 \\
0 & \cdots & 0 & 1 & 1 \\
1 & \cdots & 1 & 0 & 1 \\
\end{array}
\]

\(M_{IV}, p \geq 1\)

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 \\
\end{array}
\]

\(M_{V}\)

Approach: Use a search tree algorithm.

Repeat:

1. Search for a “forbidden submatrix”.
2. Branch on which of its columns has to be deleted.
Min-COS-C / Min-COS-R on \((\ast, \Delta)\)-Matrices

Search Tree Algorithm:

Finite size \(c\) of forbidden matrices \(\Rightarrow\) search tree of size \(O(c^k)\).
(Alternatively: Factor-\(c\) approximation algorithm.)
Min-COS-C / Min-COS-R on \((\ast, \Delta)\)-Matrices

A \((\ast, \Delta)\)-matrix can contain

- \(M_{I_p}\) with unbounded size,
- \(M_{II_p}\) with \(1 \leq p \leq \Delta - 2\),
- \(M_{III_p}\) with \(1 \leq p \leq \Delta - 1\),
- \(M_{IV}\), and \(M_{V}\).
Min-COS-C / Min-COS-R on \((\ast, \Delta)\)-Matrices

Problem: Matrices \(M_{I_p}\) of unbounded size can occur.
Min-COS-C / Min-COS-R on \((\ast, \Delta)\)-Matrices

Problem: Matrices \(M_{I_p}\) of unbounded size can occur.

Idea: First destroy all “small” forbidden submatrices (search tree algorithm), and then see what happens...
Min-COS-C / Min-COS-R on \((*, \Delta)\)-Matrices

Algorithmic framework for Min-COS-C / Min-COS-R:

1. Destroy the submatrices from

\[ X := \{ M_{I_p} | 1 \leq p \leq \Delta - 1 \} \cup \{ M_{II_p} | 1 \leq p \leq \Delta - 2 \} \]
\[ \cup \{ M_{III_p} | 1 \leq p \leq \Delta - 1 \} \cup \{ M_{IV}, M_{V} \}. \]

2. Destroy the remaining \( M_{I_p} \) (\( p \geq \Delta \)).

*Theorem:* If a \((*, \Delta)\)-matrix \( M \) contains none of the matrices in \( X \) as a submatrix, then \( M \) can be partitioned into “independent” submatrices that have the “circular ones property \((\text{Circ1P})\)”.

[Dom, Guo, Niedermeier, TAMC ’07]
FPT algorithm:
Running time:
\[(|\text{submatrix}|)^k \cdot (\text{search} + \text{"Circ1P} \rightarrow \text{C1P" time})\]

Old\(^4\):
\[(\Delta + 2)^k \cdot (n^{O(\Delta)} + n^{O(\Delta)})\] (only Min-COS-C)

New:
\[(\Delta + 2)^k \cdot (n^{O(1)} + O(\Delta mn))\]

Approximation algorithm:
Approximation factor: \(|\text{submatrix}|\)
Running time: \(k \cdot (\text{search} + \text{"Circ1P} \rightarrow \text{C1P" time})\)

\(^4\)[Dom, Guo, Niedermeier, TAMC ’07]
Structure of the Talk

- Algorithmic Framework
- From Circ1P to C1P
From Circ1P to C1P

Again:

Theorem: If a \((\ast, \Delta)\)-matrix \(M\) contains none of the matrices in \(X\) as a submatrix, then \(M\) can be partitioned into “independent” submatrices that have the “circular ones property (Circ1P)”.  
[Dom, Guo, Niedermeier, TAMC '07]
“Independent” Submatrices

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
<th>$c_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Graph Representation:**
  - Nodes: $c_1$, $r_1$, $c_4$, $c_5$, $r_4$
  - Edges: $c_1 ightarrow r_1 ightarrow c_4$, $c_5 ightarrow r_4$
  - Subsets: $c_2$, $r_2$, $c_3$, $c_6$
The Circular Ones Property (Circ1P)

A 0/1-matrix $M$ has the Circ1P if its columns can be permuted such that in each row the 1’s form a block \textit{when $M$ is wrapped around a vertical cylinder}. 
From Circ1P to C1P

C1P: 1’s blockwise after column permutations
Circ1P: 1’s blockwise on a cylinder after column permutations
strong C1P: 1’s blockwise \textit{without} column permutations
strong Circ1P: 1’s blockwise on a cylinder \textit{without} column permutations

(Circ1P/C1P means: Strong Circ1P/strong C1P can be obtained by column permutations.)
From Circ1P to C1P

We imagine the matrices as wrapped around a vertical cylinder.

Strong Circ1P:

Strong C1P:
From Circ1P to C1P

We imagine the matrices as wrapped around a vertical cylinder.

Strong Circ1P:

Strong C1P:

\[
\text{Strong C1P} = \text{Strong Circ1P} + \text{"cut"}
\]
From Circ1P to C1P

Our task:

strong Circ1P → column deletions → strong Circ1P + C1P
From Circ1P to C1P

Our task:

\[
\begin{align*}
\text{strong Circ1P} & \quad \rightarrow \quad \text{column deletions} & \quad \text{strong Circ1P} + \quad \text{C1P} \\
\end{align*}
\]

First consider this task:

\[
\begin{align*}
\text{strong Circ1P} & \quad \rightarrow \quad \text{column deletions} & \quad \text{strong Circ1P} + \quad \text{strong C1P} \\
\end{align*}
\]
From Circ1P to C1P

Our task:

\[
\begin{array}{c}
\text{strong Circ1P} \\
\rightarrow \\
\text{column deletions} \\
\text{strong Circ1P + C1P}
\end{array}
\]

First consider this task:

\[
\begin{array}{c}
\text{strong Circ1P} \\
\rightarrow \\
\text{column deletions} \\
\text{strong Circ1P + strong C1P}
\end{array}
\]

*Obs.*: Deleting a consecutive set of columns is always optimal.
From Circ1P to C1P

Our task:

\[
\text{strong Circ1P} \quad \longrightarrow \quad \text{column deletions} \quad \text{strong Circ1P} + C1P
\]

Easy task:

\[
\text{strong Circ1P} \quad \longrightarrow \quad \text{column deletions} \quad \text{strong Circ1P} + \text{strong C1P}
\]

We hope: Does “strong Circ1P + C1P” imply “strong C1P”? 
From Circ1P to C1P

*Conjecture:* If a matrix has the strong Circ1P and the C1P, then it has also the strong C1P.
**Conjecture:** If a matrix has the strong Circ1P and the C1P, then it has also the strong C1P.

Counterexample:
Conjecture: If a matrix has the strong Circ1P and the C1P, then it has also the strong C1P.

Counterexample:
From Circ1P to C1P

Conjecture: If a matrix has the strong Circ1P and the C1P, then it has also the strong C1P.

Counterexample:

New conjecture: If a matrix with $\geq 2\Delta - 1$ columns has the strong Circ1P and the C1P, then it has also the strong C1P.
To be proven: If a matrix with $\geq 2\Delta - 1$ columns has the strong Circ1P and the C1P, then it has also the strong C1P.

Very helpful:
*Theorem:* Let $M$ have the strong Circ1P. Then every column permutation that also yields the strong Circ1P can be obtained by a series of circular module reversals.

From Circ1P to C1P

To be proven: If a matrix with $\geq 2\Delta - 1$ columns has the strong Circ1P and the C1P, then it has also the strong C1P.

Very helpful:

Theorem: Let $M$ have the strong Circ1P. Then every column permutation that also yields the strong Circ1P can be obtained by a series of circular module reversals.

From Circ1P to C1P

To be proven: If a matrix with $\geq 2\Delta - 1$ columns has the strong Circ1P and the C1P, then it has also the strong C1P.

Very helpful:

Theorem: Let $M$ have the strong Circ1P. Then every column permutation that also yields the strong Circ1P can be obtained by a series of circular module reversals.

From Circ1P to C1P

To be proven: If a matrix with $\geq 2\Delta - 1$ columns has the strong Circ1P and the C1P, then it has also the strong C1P.

Very helpful:

Theorem: Let $M$ have the strong Circ1P. Then every column permutation that also yields the strong Circ1P can be obtained by a series of circular module reversals.

Now to be proven: Let $M$ be a matrix with with $\geq 2\Delta - 1$ columns that has the strong Circ1P and the strong C1P. Reversing an arbitrary circular module of $M$ does not affect these properties.
From Circ1P to C1P

Algorithm for Min-COS-C on matrices with Circ1P:

1. Permute the columns to get the strong Circ1P.
2. Search for a set of *consecutive* consecutive columns whose deletion yields the strong C1P.
Main Open Question

How can a matrix that has the (strong) Circ1P be modified by deleting a minimum number of 1-entries such that the resulting matrix has the C1P?