Connections between cutting-pattern sequencing, VLSI design, and flexible machines

Alexandre Linhares\textsuperscript{a,b,*}, Horacio Hideki Yanasse\textsuperscript{a}

\textsuperscript{a}Brazilian Space Research Institute, INPE/LAC, PO Box 515, S.J. Campos, SP 12201-970, Brazil
\textsuperscript{b}Brazilian School of Public Administration, EBAP/FGV, Praia Botafogo 190/426, Rio de Janeiro 22257-970, Brazil

Received 1 June 2000; received in revised form 1 March 2001

Abstract

The minimization of open stacks problem (MOSP) arises on the sequencing of a set of cutting patterns in order to minimize the maximum number of open stacks around the cutting saw. A previous study formulated the problem mathematically and raised a number of theoretical conjectures. In this work we deal with those conjectures. It is shown that the MOSP is NP-hard. A connection to the field of VLSI design, joining practitioners from both computer science and operations research, is established. Additional conjectures concerning the existence of simultaneous optimal solutions to related pattern-sequencing problems are also clarified.

Scope and purpose

There has been a recent surge in interest over industrial pattern-sequencing problems. The goal of these problems generally is to find a particular sequence of production patterns that minimizes production costs. An important pattern-sequencing problem arising in settings as distinct as in the sequencing of cutting stock and in factories employing flexible machines is known as the minimization of open stacks problem (MOSP). Though this problem has been recently analyzed in detail, its computational complexity remained an open question, as were some important conjectures relating it to other industrial problems, such as the minimization of order spread problem (MORP), the minimization of tool switches problem (MTSP), and the minimization of discontinuities problem (MDP). We deal with these questions, and, moreover, we show that the MOSP is deeply related to a problem in the field of VLSI layout, thus enabling a link between researchers in computer science and operations research. © 2002 Published by Elsevier Science Ltd.

Keywords: Pattern sequencing; Flexible machines; VLSI layout; Computational complexity
1. Introduction

Consider a production setting where \( J \) distinct patterns need to be cut. Each one of these patterns may contain a combination of at most \( I \) piece types. We can define the piece-pattern relationship by a \( I \times J \) binary matrix \( P = \{ p_{ij} \} \), with \( p_{ij} = 1 \) if pattern \( j \) contains piece type \( i \), and \( p_{ij} = 0 \) otherwise.

When a pattern is cut, the pieces are stored on stacks that remain fixed around the cutting saw until each stack is completed. Each stack holds only pieces of the same type and it remains open until the last pattern containing that piece type is cut. Difficulties in handling a larger number of distinct open stacks appear, for instance, if the number increases beyond the system capability, so some of the stacks must be temporarily removed, yielding additional costs, higher production time, and higher associated risks (as observed in glass-cutting settings). We are thus interested in the sequencing of the patterns to minimize the maximum number of open stacks during the cutting process (An analogous problem arises from the sequencing of tasks in a flexible machine with tooling constraints [1]). An open stacks versus cutting instants matrix \( Q^\pi = \{ q_{ij}^\pi \} \) is defined by

\[
q_{ij}^\pi = \begin{cases} 
1, & \text{if } \exists x, \exists y | \pi(x) \leq j \leq \pi(y) \text{ and } p_{ix} = p_{iy} = 1, \\
0, & \text{otherwise,}
\end{cases}
\]  

(1)

where \( \pi \) denotes a permutation of the \( \{1, 2, \ldots, J\} \) numbers, and defines a sequence on which the patterns are cut, such that \( \pi(i) \) is the order (instant) in which the \( i \)th pattern is cut. Note that matrix \( Q^\pi \) holds the consecutive-ones property for columns [2] under permutation \( \pi \): in each row, any zero between two ones will be changed to one. This is called a fill-in. Since we are interested in minimizing the number of simultaneous open stacks, we define the following cost functional:

\[
Z_{\text{MOSP}}^\pi(P) = \max_{j \in \{1, \ldots, J\}} \sum_{i=1}^{I} q_{ij}^\pi
\]  

(2)

and define the minimization of open stacks problem (MOSP) as the problem of \( \min_{\pi \in \Gamma} Z_{\text{MOSP}}^\pi(P) \), where \( \Gamma \) is the set of all possible one-to-one mappings \( \pi : \{1, 2, \ldots, J\} \to \{1, 2, \ldots, J\} \).

In Yanasse [1], a nontrivial mathematical formulation is presented, a branch and bound scheme to solve it is proposed, and a number of conjectures are raised. In this work we deal with those conjectures. Section 2 deals with the computational complexity of the MOSP, and the relations to problems such as modified cutwidth and problems in VLSI design. Section 3 considers some related pattern-sequencing problems, focusing on the possibility of solving a specific problem by approaching a related one. Section 4 places a new conjecture, based on a previously unknown relation between the minimization of tool switches problem and the minimization of discontinuities problem, which will be defined below. The conclusion discusses how our results influence the development of real applications. The first conjecture that should be dealt with concerns the computational complexity of the MOSP: is MOSP NP-hard?
2. Computational complexity of the MOSP

Proposition 1. MOSP is NP-hard.

Proof. Reduction of modified cutwidth to MOSP.

Modified cutwidth (MCUT) [3]
Instance: Graph $G = (V,E)$, positive integer $K$.
Question: Is there a one-to-one function $\pi: V \rightarrow \{1, 2, \ldots, |V|\}$ such that for all $i, 1 < i < |V|$, $|\{(u,v) \in E: \pi(u) < i < \pi(v)\}| \leq K$?

This problem asks for a linear ordering of the vertices of $G$ on a straight line such that no orthogonal plane that intersects the line will cross more than $K$ edges of $G$. (It differs from the minimum cut linear arrangement problem, which is also known as cutwidth, by the use of “<” sign instead of a “<”.) The optimization version of MCUT asks to minimize $K$ over all possible linear orderings of the vertices of $G$, i.e., $\min_{\pi \in \pi} Z_{\text{MCUT}}(G)$, where $Z_{\text{MCUT}}(G) = \max_{1 < i < |V|} |\{(u,v) \in E: \pi(u) < i < \pi(v)\}|$.

We will show how to solve an instance $G = (V,E)$ of MCUT with an algorithm for MOSP. First, construct a corresponding $I \times J$ matrix $P$ where $I = |E|$ and $J = |V|$ where we associate each edge in $E$ to a row number and each vertex of $V$ to a column number. Consider an edge $(u,v) \in E, u \in V, v \in V$. Let $i$ be the row number assigned to this edge and let $j$ and $k$ be the number of the columns assigned to vertices $u$ and $v$ respectively. Make $p_{ij} = p_{ik} = 1$, and $p_{il} = 0$ for all $l \neq j, l \neq k$.

Lemma 1. Each edge crossing in a linear layout of the vertices of $G$ corresponds to one and only one “fill-in” to problem matrix $Q^\pi$, associated to a permutation of the columns of $P$ with the consecutive-ones property for columns.

Proof. Consider an ordering $\pi$ of the vertices of $G$ and an edge $(u,v)$ crossing an orthogonal plane at vertex $i$. Let $x$ be the row in matrix $P$ associated with this edge. From the construction of $P$, we have $p_{x,\pi(u)} = p_{x,\pi(v)} = 1$, and $p_{x,y} = 0$ for all $y < \pi(u)$ and for all $y > \pi(v)$. At this row, following the column sequence that corresponds to the vertex ordering, there must be a fill-in at position (row $x$, column $i$), since the associated matrix $Q^\pi$ holds the consecutive-ones property for columns under permutation $\pi$.

Now that this basic relation is established, all that must be done is to ensure that the maximum column sum of the MOSP reflects only the fill-ins. In order to achieve this, we create a matrix $P'$ as follows:

$p'_{ij} = p_{ij}, \forall i \in \{1, 2, \ldots, I\}, \forall j \in \{1, 2, \ldots, J\}$.

Let $C = \max_{j} \sum_{i=1}^{l} p_{ij}$, and compute a gap $g = CJ - \sum_{j=1}^{l} \sum_{i=1}^{I} p_{ij}$. The rows $I+1, I+2, \ldots, I+g$ of $P'$ have only a single nonzero element equal to 1. They are distributed over the columns in rows $I+1, I+2, \ldots, I+g$ in such a way that there are exactly $C - \sum_{j=1}^{l} p_{ij}$ ones on each
column so that columns of $P'$ have column sum equal to $C$. Now, for the corresponding MOSP instance $P'$ [of dimension $(I + g) \times J$], we have:

**Lemma 2.** $Z_{\text{MOSP}}(P') = Z_{\text{MCUT}}(G) + C$.

**Proof.** Observe that each column of $P'$ has the same number of 1’s. Thus, when solving MOSP what is being minimized is the maximum column sum of fill-ins. By Lemma 1, the solution to this instance of MOSP solves the corresponding instance of MCUT. \[QED\]

Proposition 1 closes the conjecture posed earlier [1,4].

### 2.1. Relations to VLSI design

One basic characteristic of the MOSP is the consecutive-ones property: a stack is open at the moment that the first piece of each type is cut, and only closed when the last piece of that type is cut. Many combinatorial optimization problems also exhibit the consecutive-ones property. A problem that is intrinsically related to the MOSP arises on VLSI design, and is known as the gate matrix layout problem (GMLP).

A gate matrix layout circuit consists of a set of gates (vertical wires) with transistors (dots) that are used to interconnect the gates. In Fig. 1 such a set of gates before their interconnection is shown. There must be horizontal wires known as nets interconnecting all the gates that share transistors at the same position. If we model the problem as a matrix problem, we are given a binary $I \times J$ matrix $P = \{p_{ij}\}$, with $p_{ij} = 1$ if gate $j$ holds a transistor at position $i$, and $p_{ij} = 0$ otherwise, then, the implementation of the nets leads to the consecutive-ones property. The wiring between gates is represented by the ones interconnecting the leftmost and rightmost gates, hence, the resultant matrix holds the consecutive-ones property (therefore, we can use the associated matrix $Q^x$ defined previously).

One basic feature of the problem is that the sequencing of the gates does not alter the underlying logic equation implemented by the circuit. Furthermore, it is essential to sequence the gates in order to minimize the number of tracks, i.e., the number of necessary physical rows to implement the circuit, given by

$$Z_{\text{GMLP}}(P) = \max_{j \in \{1,...,J\}} \sum_{i=1}^{I} q_{ij}^x.$$  

Therefore, GMLP is the problem of $\min_{\pi \in \mathcal{P}} Z_{\text{GMLP}}(P)$. 

![Fig. 1. A set of gates and transistors of a gate matrix layout style circuit.](image)
The number of tracks is the only variable determining the overall circuit area, since the number of gates is constant. In Fig. 2 the gate numbering and number of transistors (plus interconnections) are shown above and below the gates, respectively. In Figs. 2(a) and (b) distinct gate permutations are displayed, with the corresponding variation in each column sum. Since the permutation displayed in Fig. 2(b) yields a maximum column sum of 4, it is possible to pack the layout in 4 tracks (physical rows), as shown in Fig. 2(c).

We thus have

**Proposition 2.** For any matrix $P$, the number of open stacks $Z_{\text{MOSP}}(P)$ equals the number of tracks $Z_{\text{GMLP}}(P)$.

**Proof.** Follows trivially from their definitions. \qed

Proposition 2 shows that the computation of MOSP and GMLP is equivalent. Proposition 2 is also another proof of the NP-hardness of MOSP, since GMLP is NP-hard [5–8].

A skeptical reader may argue that “there is no need for Proposition 1, as Proposition 2 alone would do the job of proving NP-hardness of MOSP”. That is true. However, there is an additional issue here. We are not only interested in the theoretical issues of computational complexity, but also in the practical issues of comprehending what kind of problem can actually be solved by an exact MOSP algorithm (such as that proposed in Yanasse [1]). Consider the following facts: The NP-hardness of GMLP has been proved by a reduction from the interval graph augmentation problem (IGAP) [5]. However, in practice, an exact algorithm for the GMLP (and hence MOSP) cannot be of use to solve IGAP because the number of columns of the GMLP matrix grows very rapidly on the size of the IGAP graph—the growth is in fact $O(n^2)$ because each column represents an edge of the graph [6]. On the other hand, the number of columns grows only linearly on the modified cutwidth problem, because each column represents a vertex of the graph. Since the number of possible solutions depends crucially on the number of columns (which are the objects that will be permuted), modified cutwidth should therefore be more amenable to solution by a MOSP algorithm than interval graph augmentation. This is our motivation for Proposition 1.

We must also point out that there is a family of problems related to MOSP that have been studied independently in the literature. In Table 1 we have collected a set of problems that
consist of, given input $\Phi$, compute a function $f(\Phi)$ that is either equal to the number of open stacks or closely related to it (plus or minus one).

2.2. Fixed parameter tractability

Gate matrix layout is an important problem in the theory of NP-completeness for the remarkable nonconstructive results obtained recently proving the existence of a polynomial-time decision algorithm for any fixed number $k$ of tracks [15]. However, the nonconstructive existence proof provided does not yield the decision algorithm (or any method to achieve it). What this result basically shows is that, for the decision problem where one is given a binary matrix $M$ and a parameter $k$ and asked whether $M$ has a gate matrix layout with $k$ or fewer tracks, the parameter $k$ constitutes an important part of the computational complexity of the problem. If the parameter $k$ is held fixed, the problem becomes theoretically tractable (but not much is changed in practical terms, since the fixed parameter algorithm is unknown). These intriguing results have led to the recently suggested theory of fixed parameter tractability (FPT) [3] that states, for instance,

**Theorem 1** (Downey and Fellows [3], Fellows and Langston [15]). $GMLP \in FPT$, that is, gate matrix layout is fixed-parameter tractable.

This result obviously implies that

**Corollary 1.** $MOSP \in FPT$, that is, the minimization of open stacks problem is fixed-parameter tractable.

If we fix the number of open stacks, then there is a polynomial-time algorithm for the decision version of MOSP. Once again, this is mainly a theoretical result, since the decision algorithms
for each distinct $k$ are unknown. We will return to the issue of fixed parameter tractability in Section 4.

3. Related pattern-sequencing problems

Other related problems studied by Yanasse [1] include the minimization of tool switches problem (MTSP), the minimization of order spread problem (MORP), and the minimization of discontinuities problem (MDP). These problems, though closely related, are not equivalent to each other or to MOSP (and GMLP).

Let us consider the related problem where there is a maximum capacity of stacks that may be placed around the saw. If a certain cutting sequence exceeds this limit capacity at some point, then one or more stacks will need to be moved, to clear some space for the subsequent processing. These stacks must then be brought back for further processing; an operation that increases the cost and the associated risks of the cutting process. Thus, it is important to minimize the number of stack switches. The minimization of stack switches has been studied in the literature of flexible machines, under the guise of minimization of tool switches problem (MTSP).

Since there is a maximum capacity $C \leq I$ of stacks, it is assumed that each pattern has at most $C$ piece types, i.e., $\sum_{i=1}^{I} p_{ij} \leq C$ for all $j = 1, 2, \ldots, J$. A cutting sequence is given by a permutation $\pi$ of the columns of $P$, leading us to matrix $M = \{m_{ij}^\pi\}$, with $m_{ij}^\pi = P_{\pi(x)}$ for $x \in \{1, 2, \ldots, J\}$, and $m_{i0}^\pi = 0$ by definition. The consecutive-ones property does not necessarily hold for $M$, for no fill-ins are performed after the columns’ permutation. We associate a binary $I \times J$ matrix $R = \{r_{ij}^\pi\}$, such that the following restrictions are met:

- $r_{ij}^\pi \geq m_{ij}^\pi$ for all $i = 1, 2, \ldots, I, j = 1, 2, \ldots, J$ (all the required stacks at the $j$th-cutting step must be present; matrix $R$ may have fill-ins, as long as it respects the following restriction).
- $\sum_{i=1}^{I} r_{ij}^\pi = C$ for all $j = 1, 2, \ldots, J$ (there are $C$ stacks at each cutting instant, it is advantageous to maintain the $C$ stacks until a switch is necessary), and $r_{i0}^\pi = 0$ for all $i$. Matrix $R$ holds the current (not removed) stacks versus cutting instants relation for the MTSP: $r_{ij}^\pi = 1$ implies and is implied by stack $i$ present at instant $j$. Using a formulation analogous to Crama et al. [16], we define the cost function

$$Z_{\text{MTSP}}(P) = \sum_{j=1}^{J} \sum_{i=1}^{I} r_{ij}^\pi (1 - r_{i,j-1}^\pi)$$

and we minimize it over the set of all possible permutations $\Gamma$, i.e., $\min_{\pi \in \Gamma} Z_{\text{MTSP}}(P)$.

Bard [17], Tang and Denardo [18] and Crama et al. [16] have previously studied this problem. Tang and Denardo [18] proposed a mathematical formulation for MTSP, and argued that MTSP is NP-hard. Unfortunately, as pointed out in Yanasse [1] and Crama et al. [16], their argument has flaws. Crama et al. [16] then reduced the problem of minimal length traveling salesman path on edge graphs to MTSP, and demonstrated that the problem is in fact NP-hard (an even simpler proof uses a reduction from consecutive blocks minimization [19, problem SR17]).
The objective of the MORP is to minimize the maximum order spread, that is, \( \min_{\pi \in \mathcal{P}} Z_{\text{MORP}}^\pi (P) \), where

\[
Z_{\text{MORP}}^\pi (P) = \max_{i \in \{1, \ldots, I\}} \left( \sum_{j=1}^{J} q_{ij}^\pi - 1 \right).
\]  

(5)

The MORP is an NP-hard problem related to bandwidth minimization [1]. Madsen [20] has tried to solve the MORP by minimizing the number of discontinuities that arise on a production sequence, that is, \( \min_{\pi \in \mathcal{P}} Z_{\text{MDP}}^\pi (P) \), where

\[
Z_{\text{MDP}}^\pi (P) = \sum_{j=1}^{J} \sum_{i=1}^{I} m_{i,j}^\pi (1 - m_{i,j-1}^\pi).
\]  

(6)

where the \( J \) columns of matrix \( M \) are obtained by a permutation of the columns of matrix \( P \), as defined previously (note that the associated matrix for the MORP has fill-ins, but not for the MDP). Madsen’s proposal does not always lead to optimal MORP solutions, for these problems—though closely related—are not equivalent. An issue that is not addressed by Madsen [20] concerns the computational complexity of the MDP. If the MDP is easily solvable then it might be of use in the development of approximation algorithms, or maybe even of fully approximation schemes for the MORP. Unfortunately, we must point out that MDP is NP-hard; it has been previously studied under the name of consecutive blocks minimization [19, Problem SR17]. Therefore, MDP is as hard to solve as MORP, and cannot be used in the development of polynomial-time approximation algorithms for MORP (also it is of limited practical use to heuristically solve MORP as suggested by Madsen [20]).

Although MORP, MDP and MTSP are closely related to the MOSP, these problems are not equivalent. However, the following conjectures were posited in Yanasse [1], in an attempt to exploit some special properties that the MOSP may have

**Conjecture 1.** There is always a solution that is simultaneously optimal for MOSP and MORP.

**Conjecture 2.** There is always a solution that is simultaneously optimal for MOSP and MDP.

**Conjecture 3.** There is always a solution that is simultaneously optimal for MORP and MDP.

**Conjecture 4.** There is always a solution that is simultaneously optimal for MOSP and MTSP.

We now turn our attention to these conjectures. Consider the following problem instances given by matrices \( P_1 \) and \( P_2 \). Table 2 presents the costs associated with each permutation of the columns of these matrices. Note that only half of the possible permutations are enumerated,
Table 2  
Objective function values of problems $P_1$ and $P_2$ according to the permutations of the columns

<table>
<thead>
<tr>
<th>Sequence</th>
<th>$Z_{MOSP}(P_1)$</th>
<th>$Z_{MORP}(P_1)$</th>
<th>$Z_{MOSP}(P_2)$</th>
<th>$Z_{MORP}(P_2)$</th>
<th>$Z_{MDP}(P_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi = {1,2,3,4}$</td>
<td>11</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>$\pi = {1,2,4,3}$</td>
<td>9</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>$\pi = {1,3,2,4}$</td>
<td>8</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>$\pi = {1,4,2,3}$</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>$\pi = {1,3,4,2}$</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>$\pi = {1,4,3,2}$</td>
<td>10</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>$\pi = {2,1,3,4}$</td>
<td>11</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>$\pi = {2,1,4,3}$</td>
<td>9</td>
<td>3</td>
<td>11</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>$\pi = {3,1,2,4}$</td>
<td>8</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>$\pi = {4,1,2,3}$</td>
<td>9</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>$\pi = {3,1,4,2}$</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>$\pi = {4,1,3,2}$</td>
<td>10</td>
<td>3</td>
<td>10</td>
<td>3</td>
<td>18</td>
</tr>
</tbody>
</table>

because these problems are symmetric (i.e., the cost of sequence $\{1,2,3,4\}$ equals that of sequence $\{4,3,2,1\}$).

\[
P_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.
\]

**Proposition 3.** Conjecture 1 is not true.

**Proof.** The conjecture does not hold for matrix $P_1$.  

**Proposition 4.** Conjecture 2 is not true.

**Proof.** The conjecture does not hold for matrix $P_2$.  

**Proposition 5.** Conjecture 3 is not true.
Proof. The conjecture does not hold for matrix $P_2$. □

**Proposition 6.** Conjecture 4 is not true.

Proof. Let an MTSP instance be defined by the parameters $(P, C)$, given by the binary $I \times J$ matrix $P$ where $\sum_{i=1}^{I} p_{ij} = \sum_{i=1}^{I} p_{ij}$ for all $j \neq k$, and a maximum capacity of stacks $C = \sum_{i=1}^{I} p_{ij}$ for any $j \in \{1, 2, \ldots, J\}$. For this special full capacity case of the MTSP, $C$ stacks have to be used at each cutting instant. This means that a stack (tool) can only be kept open (at the tool magazine) at cutting instant $j$ if it is needed in cutting instants $j - 1$ and $j$. Hence, in this particular case of the MTSP, each stack switch implies and is implied by a discontinuity, and the number of stack switches equals the number of discontinuities. Therefore, $Z_{\text{MTSP}}(P) = Z_{\text{MDP}}(P)$ for all $P$ in this special case. Hence, if there is a counterexample to Conjecture 2 based on this special case of the MDP, it follows that this very same sequence is a counterexample to Conjecture 4. Observe that $P_2$ satisfies $\sum_{i=1}^{I} p_{ij} = \sum_{i=1}^{I} p_{ik}$ for all $j \neq k$ and is a counterexample to Conjecture 2. Thus, Conjecture 4 is false.

Note, however, that Conjecture 4 is true for the special case where $C \geq \min_{x \in X} Z_{\text{MOSP}}(P)$ [1].

From the previous results and the observations made in the proof of Proposition 6 we can also conclude that:

**Proposition 7.** It is not true that there is always a solution that is simultaneously optimal for MORP and MTSP.

Proof. Consider problem instance $P_2$, Proposition 5 and the fact that MDP and MTSP are equivalent for this instance if $C = 6$. □

The following conjecture is placed and left open, based on the fact that it is true for the special full capacity case of the MTSP.

**Conjecture 5.** There is always a solution that is simultaneously optimal for MDP and MTSP.

We do not know whether these examples are minimal; we are also intrigued with the question of finding the structural properties (contained in $P_1$ and $P_2$) that would actually define why an arbitrary matrix is in fact a counterexample to the conjectures.

Another interesting complexity result that has appeared for the GMLP is the following:

**Proposition 8.** MOSP does not have an absolute approximation algorithm (unless $P = NP$).

We would like to extend this result to the following problems:

**Proposition 9.** MTSP does not have an absolute approximation algorithm (unless $P = NP$).

**Proposition 10.** MDP does not have an absolute approximation algorithm (unless $P = NP$).
We refer the reader to Deo et al. [21] for the proof of these propositions. Note that in the corresponding proof for the MTSP, the tool capacity $C$ must also be multiplied. Note also that this transformation does not hold for the MORP.

4. A new conjecture

Consider the following fact, pointed out in Yanasse [1]: MOSP and MTSP are equivalent if $C = \text{OPT}_{\text{MOSP}}(P)$. Thus, in this case, if a fixed parameter algorithm $A^k_{\text{MOSP}}(P)$ for MOSP (or for GMLP) with parameter fixed at $k$ decides YES to instance $P$, then in the corresponding MTSP with capacity $C = k$ it is possible to process all stacks in such a way that no stack removal (switch) is performed until each stack is fully completed, and hence, the associated matrix $R$ holds the consecutive ones property. Thus, the polynomial-time fixed parameter algorithms known to exist due to the work of Fellows and Langston [3,15] may be used to solve the decision version of the MTSP in these special cases. This, coupled with the fact that other related combinatorial width problems such as minimum cut linear arrangement [19, GT44] have proved fixed parameter tractable [3], leads us to place the following open conjecture:

**Conjecture 6.** $\text{MTSP} \in \text{FPT}$, that is, minimization of stack (tool) switches is fixed-parameter tractable.

Another motivation for the study of Conjecture 6 comes from the fact that

**Proposition 12.** If $\text{MTSP} \in \text{FPT}$, then $\text{MDP} \in \text{FPT}$.

**Proof.** We hypothesize a polynomial-time fixed parameter algorithm $A^k_{\text{MTSP}}(M,C)$, where $M$ is the binary matrix for the MTSP, $C$ is the stack (tool) capacity, and $k$ is the parameter in question. We must show that this implies the existence of an algorithm $A^{k'}_{\text{MDP}}(M')$ that decides in polynomial-time whether matrix $M'$ holds a column sequence with $k'$ or less discontinuities.

We thus start with an MDP instance $M' = \{m'_{ij}\}$.

Let $C = \max_j \sum_{i=1}^I m'_{ij}$, and compute a gap $G = CJ - \sum_{j=1}^{J'} \sum_{i=1}^I m'_{ij}$.

Create matrix $M$ as follows:

$$m_{ij} = m'_{ij}, \quad \forall i \in \{1,2,\ldots,I\}, \quad \forall j \in \{1,2,\ldots,J\}$$

and rows $I + 1, I + 2, \ldots, I + G$ of $M$ having only a single nonzero element equal to 1. The ones are distributed over the columns in rows $I + 1, I + 2, \ldots, I + G$ in such a way that there are exactly $C - \sum_{i=1}^I m_{ij}$ ones on each column. Hence all columns of matrix $M$ have column sum equal to $C$.

Let $k = k' + G$.

It is easy to see that the MTSP instance given by $\langle M, C \rangle$ is the full capacity special case discussed earlier, and that $\text{OPT}_{\text{MTSP}}(M,C) = \text{OPT}_{\text{MDP}}(M') + G$. Thus, $A^k_{\text{MTSP}}(M,C)$ yields YES if and only if $A^{k'}_{\text{MDP}}(M')$ also yields YES. Furthermore, this is all done in polynomial-time.
5. Summary

We presented some theoretical results for the minimization of open stacks problem and other pattern sequencing problems. The computational complexity of MOSP, previously conjectured to be NP-hard, is confirmed. Other theoretical results show that MOSP is significantly distinct from MORP, MDP, and MTSP; in the sense that not always there is a simultaneously optimal solution to both problems. Moreover, unless P = NP, there is no absolute approximation algorithm for either the MOSP or the related MDP and MTSP.

We have shown that the solution of MOSP is equivalent to solving GMLP. However, this result is restricted to the classical formulations of these problems. Researchers in VLSI design face other restrictions (which were not considered in the original formulation of GMLP), such as gates of varying sizes, or maybe the restrictions arising on the problem of gate matrix partitioning. On the other hand, there are numerous restrictions which have not been considered in the MOSP, but which could be relevant in a cutting stock setting. For instance, consider the saw time, which is a bounded resource, or the size of the stacks, or perhaps the additional constraints that must be considered if one is approaching the problems of cutting pattern generation and sequencing in an integrated manner. These examples show that, as important as the classical formulations of these problems are, they are still limited, as there are additional cost considerations which they do not address.

Other important questions remain open. For instance, the generalization of Conjecture 5 beyond the special case of full capacity, or the development of polynomial-time relative approximation algorithms (or even fully polynomial approximation schemes) still remain as open possibilities.

Conjecture 6 is also challenging. Of related interest is the question of whether a polynomial-time fixed parameter decision algorithm for MTSP is of any help to construct a fixed parameter solution (we refer the reader to Downey and Fellows [3] and Fellows and Langston [15]). The interest of this work goes clearly beyond pattern sequencing; there is a family of combinatorial problems equivalent to MOSP studied independently, as depicted on Table 1. Moreover, we note that the pattern sequencing problems considered in this paper have received increasing attention in the literature. For instance, Fink and Voss [4], Crama et al. [16], Faggioli and Bentivoglio [22], Foerster and Wäscher [23], Linhares et al. [24] and Djellab et al. [25] all deal with modern heuristics for the problems considered here.

Finally, there are a number of reasons this work should be considered by practitioners interested in solving these pattern sequencing problems, as algorithmic design is directly influenced by the results obtained here. Among the immediate implications, we may cite:

(i) Our paper fundamentes the choice made in a previous article [4] for a heuristic algorithm for the MOSP, by showing that the problem is NP-hard, and thus that a heuristic method is the best that can be reasonably expected for large-sized problems. Hopefully, knowledge that the problem is NP-hard may even stimulate practitioners to develop new, advanced, heuristics and algorithms for its solution.

(ii) As is well known, a common way to approach a combinatorial problem is by solving a related one; for example, in Madsen [20], the MDP is solved by approaching a related traveling salesman problem. The crucial fact that supports this mode of approach is that there
are solutions which are optimum for both problems. Unfortunately, our results demonstrate that this type of approach would not yield efficient algorithms for the considered problems, as there is no guarantee of simultaneously optimum solutions for many problem combinations.

(iii) Researchers may also be interested in developing absolute approximation algorithms for problems such as the MTSP: it just lies as an open, intriguing, possibility. However, the existence of such an algorithm would imply that $P = NP$, and thus such expectations are unreasonable.

(iv) Finally, as we have pointed out, the body of literature concerning GMLP should be studied by those who approach the MOSP, and vice-versa. The artificial intelligence based GM_Plan algorithm [26], for instance, can be used to sequence cutting patterns in order to solve the MOSP. Then again, the methods of Fink and Voss for the MOSP [4] can be used to design integrated circuits. These are possibilities that were not intended back then when these algorithms were developed. There are numerous additional algorithms that, as soon as the equivalence of GMLP and MOSP is acknowledged, gain an increased applicability to a new industrial domain. It is remarkable to see that a theoretical statement, such as Proposition 2, opens up new, unexplored, pathways for practitioners in both computer science and operations research.

Acknowledgements

This work has been financed by grants from the FAPESP Foundation. It has come to our notice that Prof. Carlo Alberto Bentivoglio has also derived Proposition 2 independently (personal communication, May 1999).

References


Alexandre Linhares is Professor of Logistics and Operations Management at the Brazilian School of Public Administration (EBAP) of the Getulio Vargas Foundation (FGV). He holds a Computer Science Ph.D. from the Brazilian Space Research Institute. His work has appeared in Artificial Intelligence, AI Communications, Biological Cybernetics, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, and International Journal of Modern Physics.