# Approximation of Greedy Algorithms for Max-ATSP, Maximal Compression, Maximal Cycle Cover, and Shortest Cyclic Cover of Strings 

Bastien Cazaux and Eric Rivals ${ }^{\star}$<br>L.I.R.M.M. Université Montpellier II, CNRS U.M.R. 5506<br>161 rue Ada, F-34392 Montpellier Cedex 5, France<br>\{cazaux, rivals\}@lirmm.fr


#### Abstract

Covering a directed graph by a Hamiltonian path or a set of words by a superstring belong to well studied optimisation problems that prove difficult to approximate. Indeed, the Maximum Asymmetric Travelling Salesman Problem (Max-ATSP), which asks for a Hamiltonian path of maximum weight covering a digraph, and the Shortest Superstring Problem (SSP), which, for a finite language $P:=\left\{s_{1}, \ldots, s_{p}\right\}$, searches for a string of minimal length having each input word as a substring, are both Max-SNP hard. Finding a short superstring requires to choose a permutation of words and the associated overlaps to minimise the superstring length or to maximise the compression of $P$. Hence, a strong relation exists between Max-ATSP and SSP since solving Max-ATSP on the Overlap Graph for $P$ gives a shortest superstring. Numerous works have designed algorithms that improve the approximation ratio but are increasingly complex. Often, these rely on solving the pendant problems where the cover is made of cycles instead of single path (Max-CC and SCCS). Finally, the greedy algorithm remains an attractive solution for its simplicity and ease of implementation. Its approximation ratios have been obtained by different approaches. In a seminal but complex proof, Tarhio and Ukkonen showed that it achieves $1 / 2$ compression ratio for Max-CC. Here, using the full power of subset systems, we provide a unified approach for proving simply the approximation ratio of a greedy algorithm for these four problems. Especially, our proof for Maximal Compression shows that the Monge property suffices to derive the $1 / 2$ tight bound.


## 1 Introduction

Given a set of words $P=\left\{s_{1}, \ldots, s_{p}\right\}$ over a finite alphabet, the Shortest Superstring Problem (SSP) or Maximal Compression (MC) problems ask for a s. $u$ that contains each of the given words as a substring. It is a key prc compression and in bioinformatics, where it models the question of sec bly. Indeed, sequencing machines yield only short reads that need to according to their overlaps to obtain the whole sequence of the target Recent progress in sequencing tech 1 an exponent ute the nee t assembly alg



These proble which the input overlap between Covering the O a maximum weight cycli pression or of Shortest


Table 1: The approximation perf tion problems considered here. I lines). For each problem, the best greedy approximation ratio, its tightness, and the bibliographic reference are shown. Highlighted in blue: the approximation bounds for which we provide a proof relying on subset systems. The bound for Maximum Cyclic Cover was open. "Poly" means that the problem is solvable in polynomial time. A "y" after the bound means that it is tight.

Our contributions: Subset systems were introduced rece approximation performances of greedy algorithms in a unified $f$ tioned earlier, the rat have been shown wit With subset systems, on four of these prol the results mentione cepts, we study the c we focus on the $M a$ regarding Shortest C
l Compression on one hand, as well as between Maximum Cyclic Cover nd Shortest Cyclic Cover of Strings on the other. Both Max-ATSP and ve been extensively studied as essential computer science problems. Table $l l$ these problems and their greedy approximation ratios. (in columns), while the type of corer can ve a rranmuonan path S on the Overlap Graph, in
s the asymmetric maximum
l example of overlap graph.
-eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
tion ratios. S on the Overlap Graph, in
s the asymmetric maximum
lexample of overlap graph.
-eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
tion ratios. s on the Overlap Graph, in
s the asymmetric maximum
1 example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
ion ratios. s on the Overlap Graph, in
s the asymmetric maximum
1 example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
ion ratios. S on the Overlap Graph, in
s the asymmetric maximum
l example of overlap graph.
-eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
tion ratios. S on the Overlap Graph, in
s the asymmetric maximum
l example of overlap graph.
-eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
tion ratios. S on the Overlap Graph, in
s the asymmetric maximum
t example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
ely. This expresses the rela-
man Problem (Max-ATSP)
een Maximum Cyclic Cover
ther. Both Max-ATSP and
ter science problems. Table
tion ratios. on the Overlap Graph, in
sthe asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
cly. This expresses the rela-
nan Problem (Max-ATSP)
en Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table
ion ratios. on the Overlap Graph, in
sthe asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
cly. This expresses the rela-
nan Problem (Max-ATSP)
en Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table
ion ratios. on the Overlap Graph, in
s the asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
roblems of Maximal Com-
rly. This expresses the rela-
nan Problem (Max-ATSP)
een Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table
tion ratios. on the Overlap Graph, in
s the asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
roblems of Maximal Com-
rly. This expresses the rela-
nan Problem (Max-ATSP)
een Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table
tion ratios.



$\qquad$
$\qquad$





 on the Overlap Graph, in
the asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
ly. This expresses the rela-
man Problem (Max-ATSP)
en Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table
tion ratios. on the Overlap Graph, in
the asymmetric maximum
example of overlap graph.
eight Hamiltonian path or
problems of Maximal Com-
ly. This expresses the rela-
man Problem (Max-ATSP)
en Maximum Cyclic Cover
her. Both Max-ATSP and
ter science problems. Table

tion ratios. | set o |
| ---: | :--- |
| Comi |
| Superstr |
| Cyclic C |
| gorithm |
| cted gra |
| Ban path |



Figure 1: Example of an Overlap Graph for the input words $P$ := \{baaba, babaa, aabab, babba\}.

### 1.1 Sets, strings, and overlaps.

We denote by $\#(\Lambda)$ the cardinality of any finite set $\Lambda$.
An alphabet $\Sigma$ is a finite set of letters. A linear word or string over $\Sigma$ is a finite sequence of elements of $\Sigma$. The set of all finite words over $\Sigma$ is denoted by $\Sigma^{\star}$, and $\epsilon$ denotes the empty word. For a word $x,|x|$ denotes the length of $x$. Given two words $x$ and $y$, we denote by $x y$ the concatenation of $x$ and $y$. For every $1 \leq i \leq j \leq|x|$, $x[i]$ denotes the $i$-th letter of $x$, and $x[i ; j]$ denotes the substring $x[i] x[i+1] \cdots x[j]$.

A cyclic string or necklace is a finite string in which the last symbol precedes the first one. It can be viewed as a linear string written on a torus with both ends joined.

Overlaps and agglomeration Let $s, t, u$ be three strings of $\Sigma^{\star}$. We denote by $o v(s, t)$ the maximum overlap from $s$ over $t$; let $\operatorname{pr}(s, t)$ be the prefix of $s$ such that $s=$ $\operatorname{pr}(s, t) . \operatorname{ov}(s, t)$, then we denote the agglomeration of $s$ over $t$ by $s \oplus t:=\operatorname{pr}(s, t) t$. Note that neither the overlap nor the agglomeration are symmetrical. Clearly, one has $(s \oplus t) \oplus(t \oplus u)=(s \oplus t) \oplus u$.

Example 1. Let $P:=\{a b b a a, b a a b b, a a b b a\}$. One has $o v(a b b a a, b a a b b)=b a a$ and $a b b a a \oplus b a a b b=a b b a a b b$. Considering possible agglomerations of these words, we get $w_{1}=a b b a a \oplus b a a b b \oplus a a b b a=a b b a a b b \oplus a a b b a=a b b a a b b a, w_{2}=a a b b a \oplus$ $a b b a a \oplus b a a b b=a a b b a a \oplus b a a b b=a a b b a a b b$ and $w_{3}=b a a b b \oplus a b b a a \oplus a a b b a=$ $b a a b b a a \oplus a a b b a=b a a b b a a b b a$. Thus, $\left|w_{1}\right|=|\operatorname{pr}(a b b a a, b a a b b)|+|p r(b a a b b, a a b b a)|+$ $|a a b b a|=|a b|+|b|+|a a b b a|=2+1+5=8,\|P\|-\left|w_{1}\right|=15-8=7$ and $|o v(a b b a a, b a a b b)|+|o v(b a a b b, a a b b a)|=|b a a|+|a a b b|=3+4=7$

### 1.2 Notation on graphs

We consider directed graphs with weighted arcs. A directed graph $G$ is a pair $\left(V_{G}, E_{G}\right)$ comprising a set of nodes $V_{G}$, and a set $E_{G}$ of directed edges called arcs. An arc is an ordered pair of nodes.

Let $w$ be a mapping from $E_{G}$ onto the set of non negative integers (denoted $\mathbb{N}$ ). The weighted directed graph $G:=\left(V_{G}, E_{G}, w\right)$ is a directed graph with the weights on its arcs given by $w$.

A route of $G$ is an oriented path of $G$, that is a subset of $V_{G}$ forming a chain between two nodes at its extremities. A cycle of $G$ is a route of $G$ where the same node is at both extremities. The weight of a route $r$ equals the sum of the weights of its arcs. For simplicity, we extend the mapping $w$ and let $w(r)$ denote the weight of $r$.

We investigate the performances of greedy algorithms for different types of covers of a graph, either by a route or by a set of cycles. Let $X$ be a subset of arcs of $V_{G}$. $X$ covers $G$ if and only if each vertex $v$ of $G$ is the extremity of an arc of $X$.

### 1.3 Subset systems, extension, and greedy algorithms

A greedy algorithm builds a solution set by adding selected elements from a finite universe to maximise a given measure. In other words, the solution is iteratively extended. Subset systems are useful concepts to investigate how greedy algorithms can iteratively extend a current solution to a problem. A subset system is a pair $(E, \mathcal{L})$ comprising a finite set of elements $E$, and $\mathcal{L}$ a family of subsets of $E$ satisfying two conditions:
(HS1) $\mathcal{L} \neq \emptyset$,
(HS2) If $A^{\prime} \subseteq A$ and $A \in \mathcal{L}$, then $A^{\prime} \in \mathcal{L}$. i.e., $\mathcal{L}$ is close by taking a subset.
Let $A, B \in \mathcal{L}$. One says that $B$ is an extension of $A$ if $A \subseteq B$ and $B \in \mathcal{L}$. A
 approximatio

### 1.4 Definitions of problems and related work

Graph covers Let $G:=\left(V_{G}, E_{G}, w\right)$ be a weighted directed graph.
The well known Hamiltonian path problem on $G$ requires that the cover is a single path, while the Cyclic Cover problem searches for a cover made of cycles. We consider

```
Algorithm 1: The greedy algorithm associated with the subset system ( \(E, \mathcal{L}\) )
and weight function \(w\).
    Input : \((E, \mathcal{L})\)
    The elements \(e_{i}\) of \(E\) sorted by increasing weight: \(w\left(e_{1}\right) \leq w\left(e_{2}\right) \leq \ldots \leq w\left(e_{n}\right)\)
    \(F \leftarrow \emptyset\)
    for \(i=1\) to \(n\) do
        if \(F \cup\left\{e_{i}\right\} \in \mathcal{L}\) then \(F \leftarrow F \cup\left\{e_{i}\right\}\);
        ;
    return \(F\)
```

the weighted versions of these two problems, where the solution must maximise the weight of the path or the sum of the weights of the cycles, respectively. In a general case, the graph is not symmetrical, and the weight function does not satisfy the Triangle inequality. When a Hamiltonian path is searched for, the problem is known as the Maximum Asymmetric Travelling Salesman Problem or Max-ATSP for short.

Definition 3 (Max-ATSP). Let $G$ be a weighted directed graph. Max-ATSP searches for a maximum weiaht Hamiltonian path on $G$.

Max-ATSP
and hard to app ratio of $2 / 3$ is relaxation of the ple greedy algor imation algorith shown a $1 / 3$ apl Max-ATSP is st

1 studied problem. It is known to be NP-hard x-SNP hard). The best known approximation


Definition 4 (Max Cyclic Cover). Let $G$ be a weighted directed graph. Maximum Cyclic Cover searches for a set of cycles of maximum moinht that anllontinely cover $G$.

To our knowledge, the performance of a greedy algori lic Cover (Max-CC) has not yet been established, although v h binary weights or with cycles of predefined lengths have bee

## Superstring and Maximal Compression

Definition 5 (Superstring). Let $P=\left\{s_{1}, s_{2}, \ldots\right.$ is of $\Sigma^{\star}$. $A$ superstring of $P$ is a string $s^{\prime}$ such that $s_{i}$ is a substring of $s^{\prime}$ for any $i$ in $[1, p]$.
Let us denote the sum of the lengths of the input strings by $\|S\|:=\sum_{s_{i} \in S}\left|s_{i}\right|$. For any superstring $s^{\prime}$, there exists a set $\left\{i_{1}, \ldots, i_{p}\right\}=\{1, \ldots, p\}$ such that $s^{\prime}=$ $s_{i_{1}} \oplus s_{i_{2}} \oplus \cdots \oplus s_{i_{p}}$, and then $\|S\|-\left|s^{\prime}\right|=\sum_{j=1}^{p-1}\left|\operatorname{ov}\left(s_{i_{j}}, s_{i_{j+1}}\right)\right|$.
Definition 6 (Shortest Superstring Problem (SSP)). Let $p$ be a positive integer and $P:=\left\{s_{1}, s_{2}, \ldots, s_{p}\right\}$ be a set of $p$ strings over $\Sigma$. Find $s^{\prime}$ a superstring of $P$ of minimal length.

Two approximation measures can be optimised:

- the length of the obtained superstring, that is $\left|s^{\prime}\right|$, or
- the compression of the input strings achieved by the superstring: $\|P\|-\left|s^{\prime}\right|$.

The corresponding approximation problems are termed Shortest Superstring Problem in the first case, or Maximal Compression in the second.


Thus, the ratio of approximation is $\frac{\left|w_{g}\right|}{\left|w_{o p t}\right|}=\frac{3 k+2}{3 k+1} \longrightarrow 1$ and the ratio of is $1 / 2$. In uperstring may be almost optimal in le compressi

Both
and Max-
or their $v$
Cyclic C
Shortest
and Shortest Superstrin
s, comp
e simila
gs . Th
optimal r:
For the $M$
A seminal w iteratively ur until one str spond to a g. Appendix th for an appro] s obtain

## N

ed i blen proofs are qu

Definition 8 (Shortest Cyclic Cover of Strings (SCCS)). Let $p \in \mathbb{N}$ and let $P$ be a set of $p$ linear strings over $\Sigma: P:=\left\{s_{1}, s_{2}, \ldots, s_{p}\right\}$. Find a set of cyclic strings of minimal cumulated length such that any input $s_{i}$, with $1 \leq i \leq p$, is a substring of at least one cyclic string.

Several approximation algorithms for th lem problem uses a procedure to solve SCCS on the instal polynomial time algorithm for the assignmer ification of a the importance of SCCS.

Both the Maximal Compression and the $S \quad$ igs problems can be expressed as a cover of the Overlap $G$ represent the input strings, and an arc links the vertices the ove the overlap graph is a complete graph with null or positive weights. A Hamiltonian path of this graph provides a permutation of the input strings; by agglomerating these strings in the order given by the permutation one obtains a superstring of $P$. Hence, the maximum weight Hamiltonian path induces a superstring that accumulates an optimal set of maximal overlaps, in other words a superstring that achieves maximal compression on $P$. Thus, a $\rho$ approximation for Max-ATSP gives the same ratio for Maximal Compression. The same relation exists between the Shortest Cyclic Cover of Strings and Maximum Cyclic Cover on graphs. Indeed, SCCS optimises $\|P\|-$ $\sum_{j}\left|c_{j}\right|$, where each $c_{j}$ is a cyclic string in the solution, and Max-CC optimises the cumulated weight of the cycles of $G$. With the Overlap Graph, a minimal cyclic string is associated to each graph cycle by agglomerating the input strings in this cycle. Thus, the cumulated weight of a set of graph cycles correspol by the set of induced cyclic strings. In other words, Sho could also be called the Maximal Compression Cyclic C seen as a maximisation problem). The performance of Shortest Cyclic Cover of Strings problem is declared to l saying that greedy is an exact algorithm for this probler

## 2 Maximum Asymmetric Travelling Sa Maximum Cyclic Cover Problems

Let $w$ be a mapping from $E_{G}$ onto the set of non negative integers and let $G:=$ $\left(V_{G}, E_{G}, w\right)$ be a directed graph with the weights on its arcs given by $w$. We first define a subset system for Max-ATSP and its accompanying greedy algorithm.

Definition 9. Let $\mathcal{L}_{S}$ be the powerset of $E_{G}$. We define the pair $\left(E_{G}, \mathcal{L}_{S}\right)$ such that any $F$ in $\mathcal{L}_{S}$ satisfies
(L1) $\forall x, y$ and $z \in V_{G},(x, z)$ and $(y, z) \in F$ implies $x=y$,
(L2) $\forall x, y$ and $z \in V_{G},(z, x)$ and $(z, y) \in F$ implies $x=y$.
(L3) for any $r \in N^{\star}$, there does not exis $\left.\quad, x_{1}\right)$ ) in $F$, where $\forall k \in\{1, \ldots, r\}, x_{k} \in V_{C}$

${ }^{2}$ subset $F$ of $E_{G}$, ing) arc for each for any $v \in V_{G}$, ondione changes th subset lifferent problem. As the proofs in this section do not depend in valid for these problems as well. For instance, with $r \in\{1\}$, one are forbidden; the solution is either a maximal path or


Proo,
$C$. One must show that there exists a subset $Y \subset D \backslash C$ with $\#(Y) \leq 3$ such that

 set of $F \subseteq E_{P}$ such that:
(L1) $\forall s_{i}, s_{j}$ and $s_{k} \in S, s_{i} \odot s_{k}$ and $s_{j} \odot s_{k} \in F \Rightarrow i=j$, i.e. for each string, there is only one overlap to the left
(L2) $\forall s_{i}, s_{j}$ and $s_{k} \in S, s_{k} \odot s_{i}$ and $s_{k} \odot s_{i} \in F \Rightarrow i=j, \quad$ and only one overlap to the right
(L3) for any $r \in N^{\star}$, there exists no cycle $\left(s_{i_{1}} \odot s_{i_{2}}, \ldots, s_{i_{r-1}} \odot s_{i_{r}}, s_{i_{r}} \odot s_{i_{1}}\right)$ in $F$, such that $\forall k \in\{1, \ldots, r\}, s_{i_{k}} \in S$.
For each set $F:=\left\{s_{i_{1}} \odot s_{i_{2}}, \ldots, s_{i_{p-1}} \odot s_{i_{p}}\right\}$ that is an inclusion-wise maximal element of $\mathcal{L}_{\mathcal{P}}$, we denote by $l(F)$ the superstring of $S$ obtained by agglomerating the input strings of $P$


| Bastien | eedy Approximation for Superstring and Cyclic Covers 157 |
| :---: | :---: |
| To 1 Monge | ation ratio for the greedy algorithm, we will need the o word overlaps. |
| Lemm: $\mid \operatorname{ov}\left(s_{1}, s\right.$ | $d s_{4}$ be four different words satisfying $\left\|\operatorname{ov}\left(s_{1}, s_{2}\right)\right\| \geq$ $v\left(s_{3}, s_{2}\right) \mid$. So we have : |

When for three sets $A, B, C$, we write $A \cup B \backslash C$, it means $(A \cup B) \backslash C$. Let $A \in \mathcal{L}_{\mathcal{P}}$ and let $\operatorname{Opt}(A)$ denote an extension of $A$ of maximum weight. Thus, $\operatorname{Opt}(\emptyset)$ is an element of $\mathcal{L}_{\mathcal{P}}$ of maximum weight. The next lemma follows from this definition.

Lemma 17. Let be $F \in \mathcal{L}_{\mathcal{P}}$ and $x \in E_{P}, w(\operatorname{OPT}(F \cup\{x\})) \leq w(\operatorname{OPT}(F))$.
Now we can prove a better approximation ratio.
Theorem 18. The approximation ratio of the
 Compression problem is $1 / 2$.

ratio, we revisit the proof o denote the elements in the solution $F$, and let $F_{0}:=\emptyset$, the algorithm, in other word cture of the proof is first to s ithm, the inequality $w\left(\operatorname{OPT}\left(F_{i-1}\right)\right.$ induction on the sets $F_{i}$ starting $\operatorname{OPT}\left(F_{i-1}\right)$ is an extension of $F_{i-1}$
$F_{i-1}$ and $x_{i}$, one gets $F_{i-1} \cup\left\{x_{i}\right\}$ $s_{p}$ and $s_{o}$ such that $x_{i}=s_{p} \odot s_{o}$. Like in the proof of P subset of elements of $\operatorname{OPT}\left(F_{i-1}\right) \backslash F_{i-1}$ of the form $s_{p}$ $s_{k} \odot s_{p}$ belongs to a cycle in $\operatorname{OPT}\left(F_{i-1}\right) \cup\left\{x_{i}\right\}$. Thus,

$$
\begin{aligned}
w\left(\operatorname{OPT}\left(F_{i-1}\right)\right) & =w\left(\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}\right\}\right. \\
& \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)+w\left(Y_{i}\right)-w\left(x_{i}\right)
\end{aligned}
$$

e exist ote the where $\mathcal{P}$, and

Indeed, $w\left(\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}\right\}\right) \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)$ because $\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}\right\}$ is an extension of $F_{i-1} \cup\left\{x_{i}\right\}$ and because $\operatorname{OPT}\left(F_{i}\right)$ is an extension of maximum weight of $F_{i-1} \cup\left\{x_{i}\right\}$.

Now let us show by contraposition that for any element $y \in Y_{i}, w(y) \leq w\left(x_{i}\right)$. Assume that there exists $y \in Y_{i}$ such that $w(y)>w\left(x_{i}\right)$. As $y \notin F_{i-1}, y$ has already been considered by the greedy algorithm and not incorporated in the $F$. Hence, there exists $j \leq i$ such that $F_{j} \cup\{y\} \notin \mathcal{L}_{\mathcal{P}}$, but $F_{j} \cup\{y\} \subseteq \operatorname{OPT}\left(F_{i-1}\right) \in \mathcal{L}_{\mathcal{P}}$, which is a contradiction. Thus, we obtain $w(y) \leq w\left(x_{i}\right)$ for any $y \in Y_{i}$.

Now we know that $\#\left(Y_{i}\right) \leq 3$. Let us inspect two subcases.
Case 1: $\#\left(Y_{i}\right) \leq 2$.
We have $w(Y) \geq 2 w\left(x_{i}\right)$, hence $w\left(\operatorname{OPT}\left(F_{i-1}\right)\right) \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)+w\left(x_{i}\right)$.

Case $2: \#\left(Y_{i}\right)=3$.
There exists $s_{k}$ and $s_{k^{\prime}}$ such that $s_{p} \odot s_{k^{\prime}}$ and $s_{k} \odot s_{o}$ are in $Y_{i}$. F have $w\left(x_{i}\right)+w\left(s_{k} \odot s_{k^{\prime}}\right) \geq w\left(s_{p} \odot s_{k^{\prime}}\right)+w\left(s_{k} \odot s_{o}\right)$. As $s_{p} \odot s_{k^{\prime}}$ a to $\operatorname{OPT}\left(F_{i-1}\right)$, one deduces $s_{k} \odot s_{k^{\prime}} \notin \operatorname{OPT}\left(F_{i-1}\right)$.

We get $\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\} \in \mathcal{\mathcal { L } _ { \mathcal { P } }}$. Indeed, as $Y_{i} \subseteq 0$ a right overlap of $s_{k}$, nor a left overlap of $s_{k^{\prime}}$ can belong to $\operatorname{OPT}\left(F_{i}\right.$ adding $s_{k} \odot s_{k^{\prime}}$ to $\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}\right\}$ cannot create a cycle, since o would have already existed in $\operatorname{OPT}\left(F_{i-1}\right)$. This situation is

We have $w\left(\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}\right) \leq w(\mathrm{OP}$ because $\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}$ is an extension of $\operatorname{OPT}\left(F_{i-1} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}\right)$ is a maximum weight extension As $w\left(\operatorname{OPT}\left(F_{i-1} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}\right)\right) \leq w\left(\operatorname{OPT}\left(F_{i-1} \cup\left\{x_{i}\right\}\right)\right)$,

$$
\begin{aligned}
w\left(\operatorname{OPT}\left(F_{i-1}\right)\right) & =w\left(\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}\right)+w(Y \\
& \leq w\left(\operatorname{OPT}\left(F_{i-1} \cup\left\{x_{i}, s_{k} \odot s_{k^{\prime}}\right\}\right)\right)+w\left(Y_{i}\right)- \\
& \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)+w\left(Y_{i}\right)-w\left(x_{i}\right)-w\left(s_{k} \odot s_{k^{\prime}}\right.
\end{aligned}
$$

As $Y_{i}=\left\{s_{p} \odot s_{k^{\prime}}, s_{k} \odot s_{o}, s_{k^{\prime \prime}} \odot s_{p}\right\}$, one obtains

$$
\begin{aligned}
w\left(\operatorname{OPT}\left(F_{i-1}\right)\right) & \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)-w\left(s_{k} \odot s_{k^{\prime}}\right)+w\left(Y_{i}\right)-w\left(x_{i}\right) \\
& \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)-w\left(s_{k} \odot s_{k^{\prime}}\right)+w\left(s_{p} \odot s_{k}^{\prime}\right)+w\left(s_{k} \odot s_{o}\right)+w\left(s_{k^{\prime \prime}} \odot s_{p}\right)-w\left(x_{i}\right) \\
& \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)+w\left(s_{k^{\prime \prime}} \odot s_{p}\right) \\
& \leq w\left(\operatorname{OPT}\left(F_{i}\right)\right)+w\left(x_{i}\right) .
\end{aligned}
$$

Remembering that $\operatorname{OPT}(\emptyset)$ is an optimum solution, by induction one gets

$$
\begin{aligned}
w\left(\operatorname{OPT}\left(F_{0}\right)\right) & \leq w\left(\operatorname{OPT}\left(F_{l}\right)\right)+\sum_{i=1}^{l} w\left(x_{i}\right) \\
& \leq w\left(F_{l}\right)+w\left(F_{l}\right) \\
& \leq 2 w\left(F_{l}\right)
\end{aligned}
$$

We can substitute $w\left(\operatorname{OPT}\left(F_{l}\right)\right)$ by $w\left(F_{l}\right)$ since $F_{l}$ has a maximal weight by definition. Let $s_{\text {opt }}$ be an optimal solution for Maximal Compression, $\|P\|-\left|s_{\text {opt }}\right|=w(\operatorname{OPT}(\emptyset))$. As $F_{l}$ is maximum, $l\left(F_{l}\right)$ is the superstring of $P$ output by the greedy algorithm and thus, $\|P\|-\left|l\left(F_{l}\right)\right|=w\left(F_{l}\right)$. Therefore,

$$
\frac{1}{2}\left(\|P\|-\left|s_{\text {opt }}\right|\right) \leq\|P\|-\left|l\left(F_{l}\right)\right| .
$$

Finally, we obtain the desired ratio: the greedy algorithm of the subset system achieves an approximation ratio of $1 / 2$ for the Maximal Compression problem.

### 3.1 Shortest Cyclic

A solution for MC must However, for the Shortes length are allowed. As in the pair $\left(E_{P}, \mathcal{L}_{\mathcal{C}}\right)$, where and ( $L 2$ ). A solution for tl

cles in the constructed superstring. s problem, cycles of any positive fine a subset system for SCCS as $E_{P}$ satisfying only condition (L1) ts defined as the length of maximal


Figure 3: Impossibility to create a cycle by adding $s_{k} \odot s_{k^{\prime}}$ to $\operatorname{OPT}\left(F_{i-1}\right) \backslash Y_{i} \cup\left\{x_{i}\right\}$, without having an already existing cycle in $\operatorname{OPT}\left(F_{i-1}\right)$. Since we are adding $x_{i}$ to $\operatorname{OPT}\left(F_{i-1}\right)$, we need to
overlaps is a set of cycl can see that the proof o show that the greedy a a $1 / 1$ approximation ra

Theorem 19. The gre red: $s_{k^{\prime \prime}} \odot s_{p}, s_{p} \odot s_{k^{\prime}}, s_{k} \odot s_{o}$.
input words of $P$ as substrings. One $1 / 2$ ratio for MC can be simplified to the subset system $\left(E_{P}, \mathcal{L}_{\mathcal{C}}\right)$ achieves ly solves SCCS. of Strings problem in polynomial time.

## 4 Conclusion



## References

1. M. BläSER and B. Manthey: Approximating maximum weight cycle covers in directed graphs with weights zero and one. Algorithmica, 42(2) 2005, pp. 121-139.
2. A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis: Linear approximation of shortest superstrings, in ACM Symposium on the Theory of Computing, 1991, pp. 328-336.
3. J. Gallant, D. Maier, and J. A. Storer: On finding minimal length superstrings. Journal of Computer and System Sciences, 20 1980, pp. 50-58.
4. D. Gusfield: Algorithms on Strings, Trees and Sequences, Cambridge University Press, 1997.
5. T. A. Jenkyns: The greedy travelling salesman's problem. Networks, 9(4) 1979, pp. 363-373.
6. H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko: Approximation algorithms for asymmetric tsp by decomposing directed regular multigraphs. J. of Association for Computing Machinery, 52(4) July 2005, pp. 602-626.
7. H. Kaplan and N. Shafrir: The greedy algorithm for shortest superstrings. Information Processing Letters, 93(1) 2005, pp. 13-17.
8. J. Mestre: Greedy in Approximation Algorithms, in Proceedings of 14th Annual European Symposium on Algorithms (ESA), vol. 4168 of Lecture Notes in Computer Science, Springer, 2006, pp. 528-539.
9. G. Monge: Mémoire sur la théorie des déblais et des remblais, in Mémoires de l'Académie Royale des Sciences, 1781, pp. 666-704.
10. M. Mucha: Lyndon words and short superstrings, in Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2013, pp. 958-972.
11. K. E. Paluch: Better approximation algorithms for maximum asymmetric traveling salesman and shortest superstring. CoRR, abs/1401.3670 2014.
12. C. H. Papadimitriou and K. Steiglitz: Combinatorial optimization : algorithms and complexity, Dover Publications, Inc., 2nd ed., 1998, 496 p.
13. J. Tarhio and E. Ukkonen: A greedy approximation algorithm for constructing shortest common superstrings. Theoretical Computer Sciences, 57 1988, pp. 131-145.
14. J. S. Turner: Approximation algorithms for the shortest common superstring problem. Information and Computation, 83(1) Oct. 1989, pp. 1-20.
15. M. Weinard and G. Schnitger: On the greedy superstring conjecture. SIAM Journal on Discrete Mathematics, 20(2) 2006, pp. 502-522.
 exists a permutation of the input strings, that is a set $\left\{i_{1}, \ldots, \imath_{p}\right\}=\{1, \ldots, p\}$ such that

$$
F=\left\{s_{i_{1}} \odot s_{i_{2}}, s_{i_{2}} \odot s_{i_{3}}, \ldots, s_{i_{p-1}} \odot s_{i_{p}}\right\}
$$

Proof. By the condition (L3), cycles are forbidden in $F$. Hence there exist $s_{d_{1}}, s_{x} \in S$ such that $s_{d_{1}} \odot s_{x} \in F$, and for all $s_{y} \in S, s_{y} \odot s_{d_{1}} \notin F$.

Thus, let $\left(i_{j}\right)_{j \in I}$ be the sequence of elements of $P$ such that $i_{1}=d_{1}$, for all $j \in I$ such that $j+1 \in I, s_{i_{j}} \odot s_{i_{j+1}} \in F$, and the size of $I$ is maximum. As $F$ has no cycle (condition L3), $I$ is finite; then let us denote by $t_{1}$ its largest element. We have for all $s_{y} \in P, s_{t_{1}} \odot s_{y} \notin F$. Hence, $\cup_{j \in I} i_{j}$ is the interval comprised between $s_{d_{1}}$ and $s_{t_{1}}$.

Assume that $F \backslash\left\{\cup_{j \in I} i_{j}\right\} \neq \emptyset$. We iterate the reasoning by taking the interval between $s_{d_{2}}$ and $s_{t_{2}}$ and so on until $F$ is exhausted. We obtain that $F$ is the set of intervals between $s_{d_{i}}$ and $s_{t_{i}}$. By the condition (L1) and (L2), $s_{t_{1}}$ (resp. $s_{d_{2}}$ ) is in the interval between $s_{d_{j}}$ and $s_{t_{j}} \Rightarrow j=1$ (resp. $j=2$ ). As $s_{t_{1}} \odot s_{d_{2}} \in E$, and $F \cup\left\{s_{t_{1}} \odot s_{d_{2}}\right\} \in \mathcal{L}_{\mathcal{P}}, F$ is not maximum, which contradicts our hypothesis.

We obtain that $F \backslash\left\{\cup_{j \in I} i_{j}\right\}=\emptyset$, hence the result.

For each set $F:=\left\{s_{i_{1}} \odot s_{i_{2}}, \ldots, s_{i_{p-1}} \odot s_{i_{p}}\right\}$ that is a maximal element of $\mathcal{L}_{\mathcal{P}}$ for inclusion, remind that $l(F)$ denotes the superstring of $S$ obtained by agglomerating the input strings of $P$ according to the order induced by $F$ :

$$
l(F):=s_{i_{1}} \oplus s_{i_{2}} \oplus \cdots \oplus s_{i_{p}} .
$$

The algorithm greedy takes from set $P$ two words $u$ and $v$ having the largest maximum overlap, replaces $u$ and $v$ with $a \oplus b$ in $P$, and iterates until $P$ is a singleton.

Proposition 21. Let $F$ be the output of the greedy algorithm of subset system $\left(E_{P}, \mathcal{L}_{\mathcal{P}}\right)$, and $S$ the output of Algorithm Greedy for the input $P$. Then $S=\{l(F)\}$.

Proof. First, see that for any $i$ between 1 and $p$, there exists $s_{j}$ and $s_{k}$ such that $e_{i}=s_{j} \odot s_{k}$. If $F \cup\left\{e_{i}\right\} \in \mathcal{L}_{\mathcal{P}}$, then by Conditions (L1) and (L2), one forbids any other left overlap of $s$ As cycles are forbidde by exchanging the pa

The algorithm gr system $\left(E_{P}, \mathcal{L}_{\mathcal{P}}\right)$. By Maximal Compressio

