Covering Pairs in Directed Acyclic Graphs

Niko Beerenwinkel Stefano Beretta Paola Bonizzoni
Riccardo Dondi **Yuri Pirola**

DISCo, Univ. degli Studi di Milano-Bicocca, Milan, Italy
`yuri.pirola@disco.unimib.it`

ICTCS 2013
14th Italian Conference on Theoretical Computer Science
Palermo, September 9–11, 2013

Outline

Minimum Path Cover on DAGs

Problem: Min Path Cover on DAGs (MinPC)

Instance: a DAG $D = (N, A)$
Solution: a set Π of paths that “cover” N
Measure: $|\Pi|$

It can be solved in time $O(n^3)$
(Dilworth 1950, Fulkerson 1965, Hopcroft and Karp 1973)

Existence of fixed source and target vertices can be safely assumed

Minimum Path Cover on DAGs

MinPC has been used to solve Bioinformatics problems:

- Viral haplotype assembly (Eriksson *et al.* 2008)
- Transcript reconstruction (Trapnell *et al.* 2010)

Different applications, same problem:
Reconstructing a set of complete sequences starting from their fragments

Basic idea:

- vertices=fragments
- paths=possible complete sequences

Minimum Path Cover on DAGs

Common issue: how to choose among same-size covers?

Paired-end reads could help!

Two paired-end reads are extracted from the same sequence.

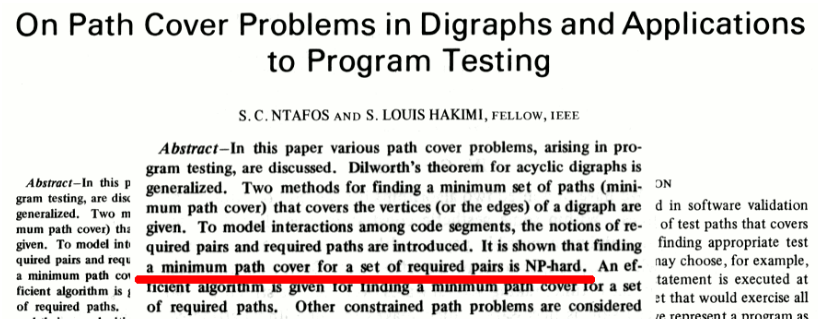
$\Rightarrow$ they must be covered by the same path

Required pair $[u, v]$:
There must exists a path in the solution that contains **both** u **and** v

Constrained MinPC

Problem: Min Path Cover with Required Pairs (MinPCRP)

Instance: a DAG $D = (N, A)$ **and** a set R of required pairs
Solution: a set Π of paths that "cover" N **and** R
Measure: $|\Pi|$



On Path Cover Problems in Digraphs and Applications to Program Testing

S. C. NTAFOS AND S. LOUIS HAKIMI, FELLOW, IEEE

Abstract—In this paper various path cover problems, arising in program testing, are discussed. Dilworth's theorem for acyclic digraphs is generalized. Two methods for finding a minimum set of paths (minimum path cover) that covers the vertices (or the edges) of a digraph are given. To model interactions among code segments, the notions of required pairs and required paths are introduced. It is shown that finding a minimum path cover for a set of required pairs is NP-hard. An efficient algorithm is given for finding a minimum path cover for a set of required paths. Other constrained path problems are considered

MaxRPSP

Possible greedy heuristic approach:
Add at each step the path that covers the maximum number of required pairs.

Problem: Max Required Pairs with a Single Path (MaxRPSP)

Instance: a DAG $D = (N, A)$ **and** a set R of required pairs
Solution: a path π
Measure: no. of required pairs covered by π

Computational complexity?

1 – MaxRPSP parameterized by the optimum

Theorem

k-RPSP is W*[1]-hard when parameterized by the number k of covered required pairs.*

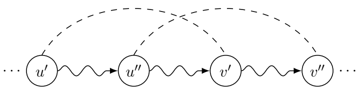
Proof (idea):
By parameterized reduction from h-Clique
(which is W[1]-hard, Downey and Fellows 1995)

Corollary: no $O(2^k P(n))$ exact algorithm exists (unless P $=$ NP)

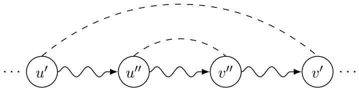
2 – Fixed-parameter algorithm

MaxRPSP has a fixed-parameter algorithm when parameterized by the *maximum number of overlapping required pairs*.

Overlapping required pairs

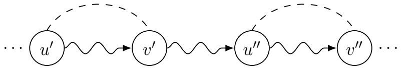


Alternated



Nested

Non-overlapping required pairs



2 – Fixed-parameter algorithm – Idea

Dynamic programming recurrence

$P\Big[\ [v_i^1, v_i^2]\ ,\ \ S\ \Big]$	Maximum number of required pairs covered by a path ending in v_i^2 and containing all the vertices in S

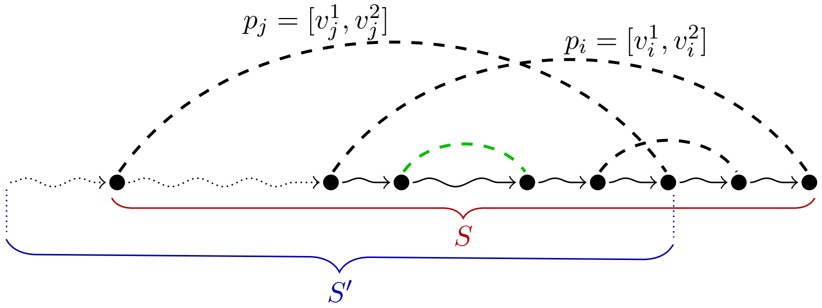
Let π be a path that:

- ends in vertex v_i^2 and covers the required pair $p_i = [v_i^1, v_i^2]$;
- contains the vertices of set S.

Suppose that $p_j = [v_j^1, v_j^2]$ is the "rightmost" required pair covered by π which is not nested in p_i.

2 – Fixed-parameter algorithm – Recursive step

$p_j =$ "rightmost" required pair covered by π and not nested in p_i



2 – Fixed-parameter algorithm – Idea

Running time: $O(4^{2h} n^2)$

- n no. of vertices
- h maximum no. of overlapping pairs

Why? Cardinality of S is bounded by $2h$!

Conclusions

- Adding constraints to Min Path Cover could help finding “better” (=closer to the hidden truth) solutions
- Various constrained variants of Min Path Cover appear to be computationally hard
- We need “good” algorithms (=approximation/fixed-parameter/heuristics/. . .)

Additional Content

Minimum Path Cover on DAGs

Problem: Min Path Cover on DAGs (MinPC)

Instance: a DAG $D = (N, A)$
Solution: a set Π of paths that "cover" N
Measure: $|\Pi|$

Algorithm (idea):

- The size of the cover is equal to the size of a maximum matching of a "corresponding" *bipartite* graph
- Maximum matching can be solved in time $O(n^3)$

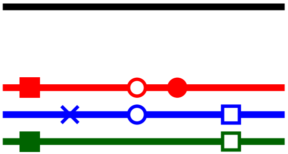
(Dilworth 1950, Fulkerson 1965, Hopcroft and Karp 1973)

Applications of MinPC in Bioinformatics

Viral quasi-species assembly

Viral reference genome

Viral quasi-species haplotypes (*unknown*)



(Eriksson *et al.* 2008)

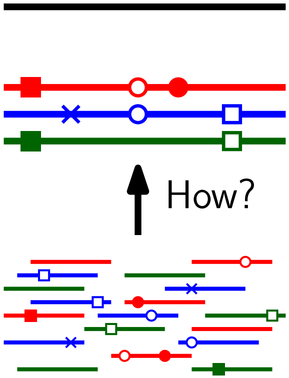
Applications of MinPC in Bioinformatics

Viral quasi-species assembly

Viral reference genome

Viral quasi-species haplotypes (*unknown*)

Sequenced fragments (*provenience unknown*)



(Eriksson *et al.* 2008)

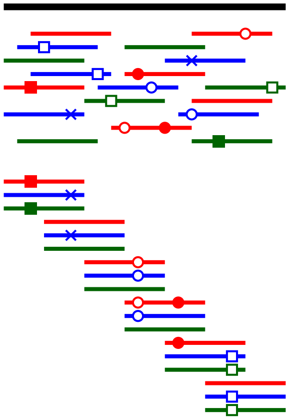
Applications of MinPC in Bioinformatics

Viral quasi-species assembly

Viral reference genome

Sequenced fragments
(*provenience unknown*)

Alignments against the
reference genome



(Eriksson *et al.* 2008)

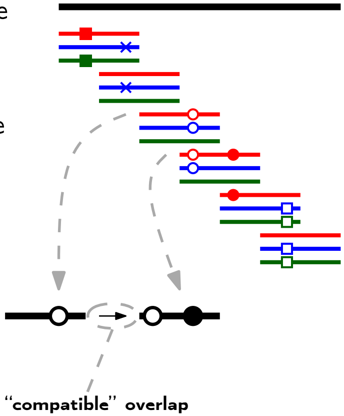
Viral quasi-species assembly – Overlap graph

Overlap graph

Viral reference genome

Alignments against the reference genome

Overlap graph



(Eriksson *et al.* 2008)

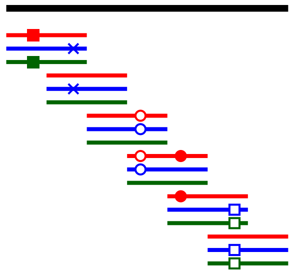
Viral quasi-species assembly – Overlap graph

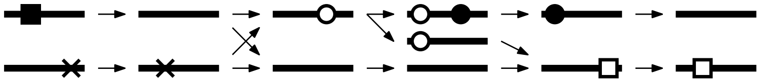
Overlap graph

Viral reference genome

Alignments against the reference genome

Overlap graph





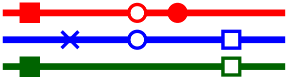
(Eriksson *et al.* 2008)

Viral quasi-species assembly – Overlap graph

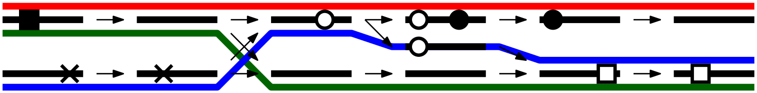
Overlap graph

Viral reference genome

Viral quasi-species haplotypes (*unknown*)



Overlap graph



Paths are putative haplotypes

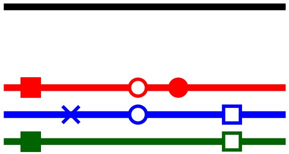
(Eriksson *et al.* 2008)

Viral quasi-species assembly – Overlap graph

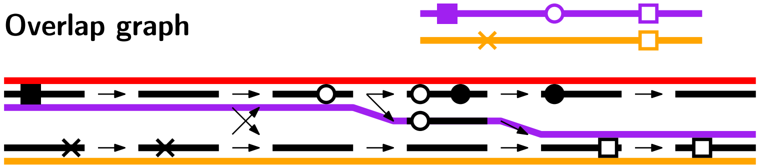
Overlap graph

Viral reference genome

Viral quasi-species haplotypes (*unknown*)



Overlap graph



Not all paths are correct!

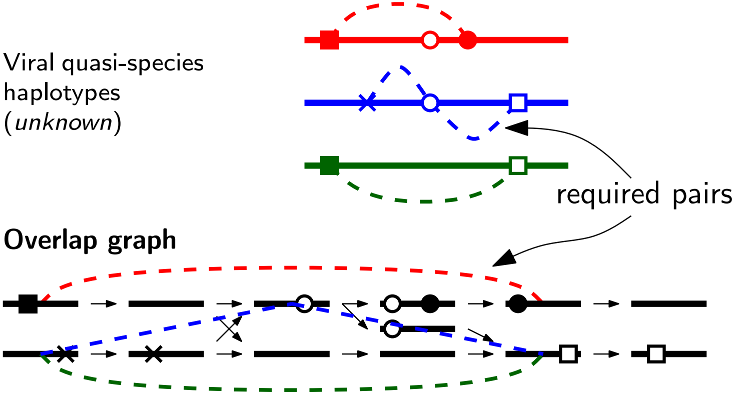
(Eriksson *et al.* 2008)

Constraining MinPC

Required pair $[u, v]$:
There must exists a path in the solution that contains u **and** v

Viral quasi-species haplotypes (*unknown*)

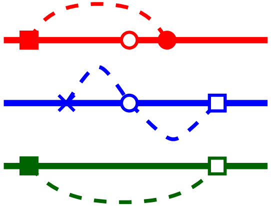
Overlap graph



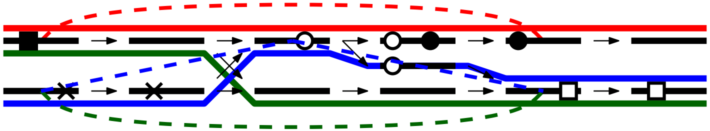
Constraining MinPC

Required pair $[u, v]$:
There must exists a path in the solution that contains u **and** v

Viral quasi-species haplotypes (*unknown*)



Overlap graph



2 – Fixed-parameter algorithm – DP recurrence

Dynamic programming recurrence

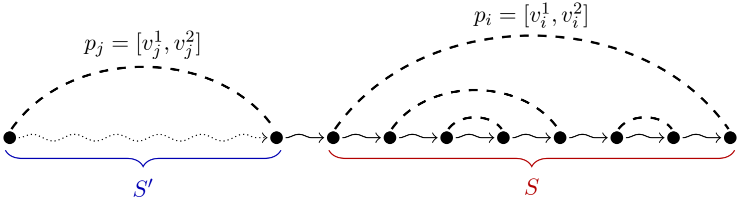
$$P\Big[[v_i^1, v_i^2], S\Big] = \max\Big\{P\Big[[v_j^1, v_j^2], S'\Big] + |Ov\Big([v_i^1, v_i^2], S \setminus S'\Big)|\Big\}$$

where:

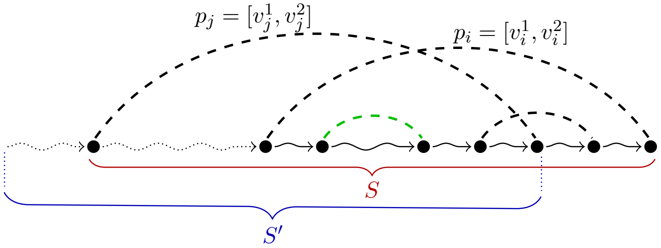
- $[v_j^1, v_j^2]$ is not nested in $[v_i^1, v_i^2]$ and $j < i$
- S' in agreement with S
- there exists a path from v_j^2 to v_i^2 containing all vertices in $S \setminus S'$
- $Ov([v_i^1, v_i^2], S \setminus S') = \{[v_h^1, v_h^2] \mid [v_h^1, v_h^2]$ nested in $[v_i^1, v_i^2] \wedge$
  $v_h^1 \in S \wedge v_h^2 \in S \setminus S'\}$

2 – Fixed-parameter algorithm – Recursive step

Case 1) p_j and p_i do not overlap:



Case 2) p_j and p_i overlap:



2 – Fixed-parameter algorithm – Idea

Running time: $O(4^{2h} n^2)$

- n no. of vertices
- h maximum no. of overlapping pairs

Cardinality of S is bounded!

Observation: for each req. pair $[v_i^1, v_i^2]$, only the vertices of required pairs overlapping $[v_i^1, v_i^2]$ really matter.

$OP\Big([v_i^1, v_i^2]\Big) =$ vertices of required pairs overlapping $[v_i^1, v_i^2]$

$$\Rightarrow |S| = O\Big(\Big|OP([v_i^1, v_i^2])\Big|\Big) = O(h)$$