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

LATA 2014
8th Int. Conf. on Language and Automata Theory and Applications
Madrid, March 10–14, 2014

Outline

1. Path Cover and *Constrained* Path Cover in Bioinformatics
2. *Our contributions:*
   - **MinPCRP** $\rightarrow$ tractability borderline
   - **MaxRPSP** $\rightarrow$ parameterized complexity
3. Conclusions and Open Problems

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)

Minimum Path Cover on DAGs

MinPC has been used to solve some problems in bioinformatics:

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

Different applications, same computational 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?

Current sequencing technologies allow to detect **pairs of fragments** that originated **from the same complete sequence**.

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

Constrained Path Cover

Two natural constrained variants of 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: $|\Pi|$

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

Constrained Path Cover – Related Works

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-5, NO. 5, SEPTEMBER 1979

On Path Cover Problems in Digraphs and Applications to Program Testing

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

- does not ask to cover all vertices (only required pairs, **MinRPC**)
- NP-hardness (with unbounded no. of paths)

MinRPC easily reduces to MinPCRP $\Rightarrow$ MinPCRP is NP-hard
(by "contracting" vertices not in required pairs)

1 – Min Path Cover with Required Pairs

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: $|\Pi|$

k-PCRP: deciding if there exists a cover with k paths

Our contributions:

- 3-PCRP is NP-complete
- 2-PCRP has a polynomial-time algorithm

1a – NP-completeness of 3-PCRP

3-PCRP is NP-complete.

Proof (idea):
By reduction from **3-coloring**
(which is NP-complete, Garey and Johnson 1979)

Corollary:
no $O(n^{f(k)})$ exact algorithm likely exists

1b – Polynomial-time algorithm for 2-PCRP

Reduction to 2-**coloring** of a graph $G = (V, E)$

$$V := R$$
$$E := \{\{r', r''\} \mid r', r'' \text{ cannot be covered by the same path}\}$$

(i.e., "incompatible" required pairs)

Proof (idea):
On the complement graph $\hat{G}$ a 2-coloring is a 2-clique partition, and a clique can be covered by a single path.

We assume all vertices belong to some req. pair, otherwise add fictitious pairs.

2 – Max Required Pairs with a Single Path

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

k-RPSP: deciding if there exists a path covering k required pairs

Our contributions:

- W[1]-hardness of k-RPSP with parameter k
- FPT algorithm with parameter *maximum overlapping degree*

2a – k-RPSP is W[1]-hard if parameterized by k

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

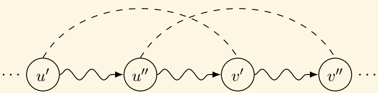
Proof (idea):
By parameterized reduction from h-Clique
(which is W[1]-hard, Downey and Fellows 1995)

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

2b – FPT algorithm for MaxRPSP

MaxRPSP has a fixed-parameter algorithm when parameterized by the **maximum overlapping degree**.

Overlapping required pairs:



Alternated

Nested

Overlapping degree of $[u, v]$**:** no. of req. pairs overlapping $[u, v]$.

2b – FPT algorithm for MaxRPSP – Idea

Dynamic programming recurrence

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

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

- n no. of vertices
- p maximum *overlapping degree*

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

(For each req. pair $[v_i^1, v_i^2]$, only vertices of required pairs overlapping $[v_i^1, v_i^2]$ really matter.)

Conclusions and Open Problems

- Adding constraints to Min Path Cover could help finding “better” (=closer to the hidden truth) solutions...
- ...but various constrained variants of Min Path Cover appear to be computationally hard
- **Open problem:** find “good” algorithms

  (*e.g.*, constant-factor approximation for MinPCRP/MaxRPSP)