PACE 2024: OCMu64, reductions a
One-sided Crossi

Ragnar Groot Koerkamp[1]

[1]ETH Zurich, @curious_coding

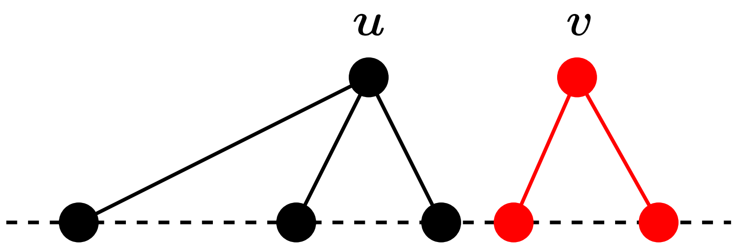
One-sided crossing minimization

Given is a bipartite graph (A, B) that is drawn in the plane at points $(i, 0)$ and $(j, 1)$ for components A and B respectively. The ordering of A is fixed. The goal is the find an ordering of B that minimizes the number of crossings when edges are drawn as straight lines.

We use $<$ to compare vertices in A in their fixed ordering. For $u \in B$, we write $N(u) \subseteq A$ for the set of its neighbours.

The **crossing number** $c(u, v)$ is $\sum_{(a,b)\in A^2} u(a)v(b)[a > b]$, the number of crossings between edges incident to u and v when u is drawn before v. For $X, Y \subseteq B$ we set $c(X, Y) = \sum_{x\in X} \sum_{y\in Y} c(x, y)$ for the cost of ordering all vertices of X before all vertices of Y. More generally, $c(X, Y, Z) = c(X, Y) + c(X, Z) + c(Y, Z)$. We also consider the *reduced cost* $r(X, Y) = c(X, Y) - c(Y, X)$, which is negative when X is better *before* Y and positive when X is better *after* Y.

We write $u \prec v$ when u must come before v in *all* minimal solutions, and say that (u, v) is a **fixed pair**. We try to fix as many pairs as possible to speed up the search. A well known result is that when $\max N(u) < \min N(v)$, then $u \prec v$.



Strongly fixed pairs

We call $u, v \in B$ a **strongly fixed pair** if $N(u) \neq N(v)$ and their degrees are n and m and for all $0 < i < n$

and a branch-and-bound solver for sing Minimization

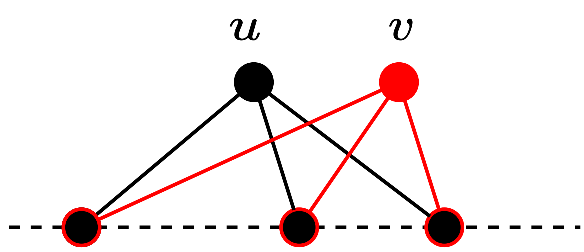
amp [1] Mees de vries [2]

[2]Unaffiliated, The Netherlands

Gluing

When u and v are consecutive in *all* optimal solutions, we **glue** them together and treat them as a single vertex.

Identical nodes: When u and v have the same neighbours, we glue them.



Greedy gluing: When $r(u,x) \leq 0$ for all $x \in B$, there is a solution that starts with u.

Practical gluing lemma (unused): Let u and v satisfy $r(u,v) \leq 0$. A non-empty subset $X \subseteq B - \{u,v\}$ is *blocking* when $c(u,X,v) \leq \min(c(u,v,X), c(X,u,v))$. If there is no blocking set, then we can glue u, v.

Again such sets X can be found or proven to not exist using a knapsack algorithm: add points $P_x = (r(u,x), r(x,v))$ and search for a non-empty set summing to $\leq (0,0)$.

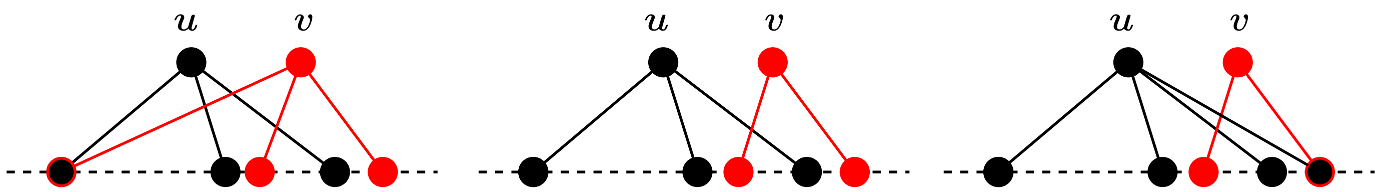
Solver

Our solver `OCMu64` uses a standard **branch-and-bound on the ordering of the solution**. We start with fixed prefix $P = ()$ and tail $T = B$, and in each step we try (a subset of) all vertices in T as the next vertex appended to P. In a preprocessing step we compute the trivial lower bound $S_0 = \sum_{u,v} \min(c(u,v), c(v,u))$ on the score.

We call $u, v \in B$ a **strongly fixed pair** if $N(u) \neq N(v)$ and their degrees are n and m and for all $0 \leq i < n$,

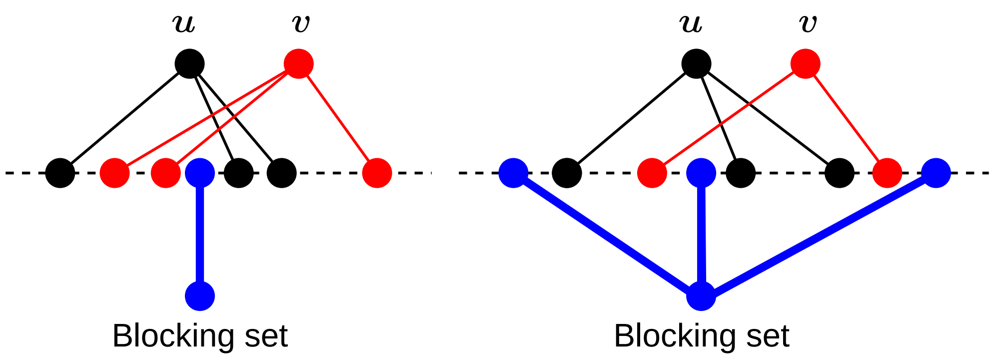
$$N(u)_i \leq N(v)_{\lfloor i \cdot m/n \rfloor}.$$

That is: for all x, *the x%'th neighbour of u is left of the x%'th neighbour of v.*



Main result: If (u, v) is strongly fixed, then $u \prec v$.

The lemma is optimal: Without further assumptions on $(A, <)$ or on the other elements of B, it is not possible to prove $u \prec v$ if (u, v) is not strongly fixed. Suppose that $u, v \in B$ with $N(u) \neq N(v)$ are not strongly fixed, and let i be such that $b := N(u)_i > N(v)_{\lfloor i \cdot m/n \rfloor} =: a$. Now take sets of vertices $L, M, R \subseteq A$ with $a < M < b$, with $L < \min(N(v))$ and with $\max(N(u)) < R$. Add a node $x \in B$ connected to L, M, R. With the correct numbers of vertices in L, M, R we can make it so that $c(v, x, u) < \min(c(x, u, v), c(u, v, x))$, so that v has to go before u.



Practically fixed pairs

Although such a node x may exist in theory, it does not have to exist in the actual set B, motivating a more practical approach:

Blocking set: Suppose $r(u, v) < 0$, i.e., u wants to go left of v. A *blocking set* $X \subseteq B - \{u, v\}$ is a set such that $c(v, X, u) \leq \min(c(v, u, X), (X, v, u))$. If there is no blocking set for (u, v), we call it a *practically fixed pair*, and $u \prec v$.

Finding blocking sets: Such a set X can be found, if one exists, using a knapsack-like algorithm: for each $x \in B - \{u, v\}$, add a point $P_x = (r(v, x), r(x, u))$, and search for a subset summing to $\leq (r(u, v), r(u, v))$.

$S_0 = \sum_{u,v} \min(c(u,v), c(v,u))$ on the score.

Graph simplification The initial solution uses the median heuristic followed by a simple local search that tries to move slices and optimally insert them. We then re-label all nodes in order to make memory accesses more efficient.

Fixed pairs We find all strongly fixed pairs and store them. For the exact track we also find practically fixed pairs. For each tail we search for *tail-local* practically fixed pairs. In each state, we only try vertices $u \in T$ not fixed after another $v \in T$.

Gluing We use the greedy gluing strategy. Our implementation of practical gluing contained a bug, so we did not use this.

Tail cache In each step, we search for the longest suffix of T that was seen before, and reuse (the lower bound on) its score. We also cache the tail-local practically fixed pairs.

Optimal insert Instead of simply appending u to P, we insert it in the optimal position. The implementation is tricky because it interacts in complicated ways with the caching of results for each tail.

Performance

Parameterized track: All 200 instances in **10.37s** total time (second place).

Exact track: 157/200 instanes.