Tight Upper Bound on the Number of Optimal Paths in Weighted Coloured-Edge Graphs

Andrew Ensor   Felipe Lillo

School of Computing and Mathematical Sciences
AUT University
Auckland. New Zealand.

24th European Conference on Operations Research, Lisbon

Outline

Multimodal Networks

- A *multimodal network* (MMN) is a networked system that has multiple *modes* of transport available.
- Examples include logistic networks, biomedical phenomena, manufacturing processes, telecommunication networks.
- Not much theory specifically developed for handling MMN.

Previous Approaches

Mathematical programming
: linear (integer, mixed integer) or non-linear (mostly second order), represent MMN by a set of equations, modes are additional decision variable indices, relaxation or cutting plane techniques used to make problem tractable.

Weighted graph
: nodes represent locations, edges represent transportation links, edge weights represent *cost*, often require edge reduction techniques to make analysis tractable.

Multi-weighted graph
: assigns multiple weights to each edge (such as cost and time), utilized for Multicriteria Shortest Path Problem, not often used for MMN but similar goals, constraints applied to make problem tractable.

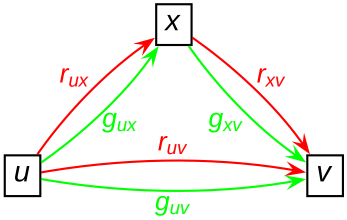
Summary of Previous Approaches

- Approaches all adapted from unimodal networks.
- Most approaches are heavily application-specific.
- Modes essentially used for constraints.
- Require heuristics or application-specific constraints to obtain tractable problems.
- Remove multimodal traits of network during analysis.

Weighted Coloured-Edge Graphs

- Idea: develop a general tool for any MMN.
- Use a graph but keep the modes throughout modelling and analysis eg for single-source shortest paths problem.
- A *weighted coloured-edge graph* $\mathbf{G} = \langle V, E, \omega, \lambda \rangle$ consists of a directed multigraph $\langle V, E \rangle$ with vertex set V and edge set E, a *weight* function $\omega \colon E \to \mathbb{R}^+$, and a *colour* function $\lambda \colon E \to M$, where M is set of possible colours for edges.
- Path weights are k-tuples, sum the path weights in each mode separately, where $k = |M|$.
- Added complexity: now the path weights are partially ordered instead of linearly ordered, get Pareto set of minimal path weights.

Example: graph for $k = 2$ and $n = 3$

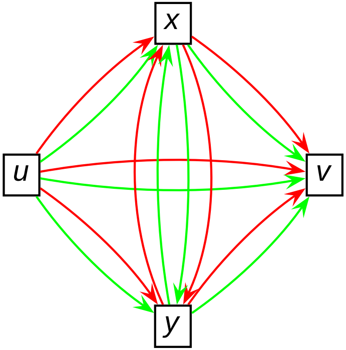


Up to six edges to consider.
Six paths from u to v:

$$(r_{ux}+r_{xv}, 0)$$
$$(r_{ux}, g_{xv})$$
$$(r_{xv}, g_{ux})$$
$$(0, g_{ux}+g_{xv})$$
$$(r_{uv}, 0)$$
$$(0, g_{uv})$$

Either two, three, or four minimal path weights depending on edge weights.

Example: graph for $k = 2$ and $n = 4$



Up to 14 edges to consider.
26 paths from u to v:

$$(r_{ux}+r_{xy}+r_{yv}, 0)$$
$$(r_{ux}+r_{xv}, 0)$$
$$(r_{ux}+r_{xy}, g_{yv})$$
$$\vdots$$
$$(r_{uv}, 0)$$
$$(0, g_{uv})$$

Between two and (after some effort) eight minimal path weights depending on edge weights.

Potential Applications

- Useful for applications where objective(s) expressible as k-ary increasing function(s) of total weights in each mode.
- Weights could be anything, eg abstract measure of *distance*, can include eg mode changes as another colour.
- Transportation networks: use distances as weights, cost/time/carbon emissions are functions of distance in each mode, post-optimal analysis possible eg how much petrol/diesel costs can alter before best path changes?
- Computer networks: use protocols as colours, can route using combinations of message-specific requirements eg secure, reliable.
- Some analogies with multicriteria problems, can apply multicriteria or brute force to the resulting Pareto set.

Potential Difficulties

- For $n = |V|$ and $k = |M|$ there are up to

$$\sum_{j=0}^{n-2} \binom{n-2}{j} k^{j+1} j!$$

paths from u to v, so factorial order $O\left(k^{n-1}(n-2)!\right)$.

- To be practical need efficient algorithm to find Pareto set.
- To be practical need the Pareto minimal paths to have manageable cardinality.
- Appears discouraging...

Surprises

- To be practical need efficient algorithm to find Pareto set.
  - Can use eg a partial-order generalisation of Dijkstra's algorithm.
  - Big Surprise: Order typically low-order polynomial in n, can use with networks as large as found in any MMN literature.
  - Big Surprise: No problem dealing with dense networks, so no need for application-specific reductions.
  - Algorithm performance depends more on k than on n or the number of edges.
- To be practical need the Pareto minimal paths to have manageable cardinality.
  - Cardinality can be *moderately bad* in theory (this talk).
  - Big Surprise: Cardinality cannot be *very bad* (this talk).
  - Big Surprise: Cardinality is typically subpolynomial in n.

Surprises

- To be practical need efficient algorithm to find Pareto set.
  - Can use eg a partial-order generalisation of Dijkstra's algorithm.
  - Big Surprise: Order typically low-order polynomial in n, can use with networks as large as found in any MMN literature.
  - Big Surprise: No problem dealing with dense networks, so no need for application-specific reductions.
  - Algorithm performance depends more on k than on n or the number of edges.
- To be practical need the Pareto minimal paths to have manageable cardinality.
  - Cardinality can be *moderately bad* in theory (this talk).
  - Big Surprise: Cardinality cannot be *very bad* (this talk).
  - Big Surprise: Cardinality is typically subpolynomial in n.

Surprises

- To be practical need efficient algorithm to find Pareto set.
  - Can use eg a partial-order generalisation of Dijkstra's algorithm.
  - Big Surprise: Order typically low-order polynomial in n, can use with networks as large as found in any MMN literature.
  - Big Surprise: No problem dealing with dense networks, so no need for application-specific reductions.
  - Algorithm performance depends more on k than on n or the number of edges.
- To be practical need the Pareto minimal paths to have manageable cardinality.
  - Cardinality can be *moderately bad* in theory (this talk).
  - Big Surprise: Cardinality cannot be *very bad* (this talk).
  - Big Surprise: Cardinality is typically subpolynomial in n.

Surprises

- To be practical need efficient algorithm to find Pareto set.
  - Can use eg a partial-order generalisation of Dijkstra's algorithm.
  - Big Surprise: Order typically low-order polynomial in n, can use with networks as large as found in any MMN literature.
  - Big Surprise: No problem dealing with dense networks, so no need for application-specific reductions.
  - Algorithm performance depends more on k than on n or the number of edges.
- To be practical need the Pareto minimal paths to have manageable cardinality.
  - Cardinality can be *moderately bad* in theory (this talk).
  - Big Surprise: Cardinality cannot be *very bad* (this talk).
  - Big Surprise: Cardinality is typically subpolynomial in n.

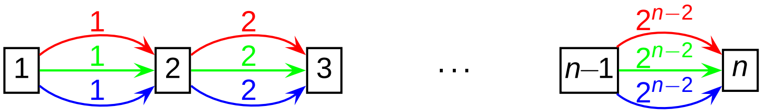
Chains

- Call a weighted coloured-edge graph a *chain* if vertices enumerated $1, 2, 3, \ldots, n-1, n$ and only have edges c_{xy} for $y = x + 1$.

$$\begin{tikzcd}
\boxed{1} \arrow[r, bend left, "r_1"] \arrow[r, "g_1"] \arrow[r, bend right, "b_1"'] & \boxed{2} \arrow[r, bend left, "r_2"] \arrow[r, "g_2"] \arrow[r, bend right, "b_2"'] & \boxed{3} & \cdots & \boxed{n-1} \arrow[r, bend left, "r_{n-1}"] \arrow[r, "g_{n-1}"] \arrow[r, bend right, "b_{n-1}"'] & \boxed{n}
\end{tikzcd}$$

- A chain has k^{n-1} paths from $u = 1$ to $v = n$.

Example: chain with k colours and n vertices



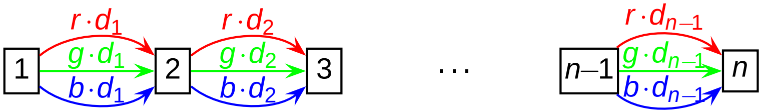
Paths from $u = 1$ to $v = n$ (if $k = 3$):

$$\begin{gathered}(2^{n-1}-1, 0, 0)\\(2^{n-1}-2, 0, 1)\\(2^{n-1}-2, 1, 0)\\\vdots\\(0, 0, 2^{n-1}-1)\end{gathered}$$

Easily show by induction on n that all k^{n-1} paths are minimal.

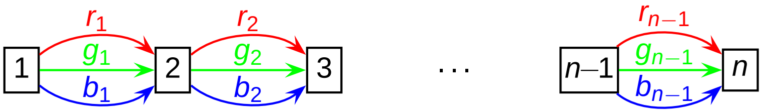
Slightly More Generally

Presume $r, g, b, \ldots$ and $d_1, d_2, \ldots, d_{n-1}$ are positive real numbers where the sum function Σ on subsets of $D = \{d_1, d_2, \ldots, d_{n-1}\}$ is one to one.



- Again all k^{n-1} paths are minimal.
- Criterion almost always true if $d_1, d_2, \ldots, d_{n-1}$ are taken from a continuous distribution.
- Conclusion: Need some independence or randomness between the edge weights from each x to $x+1$ in applications to avoid worst case scenario.

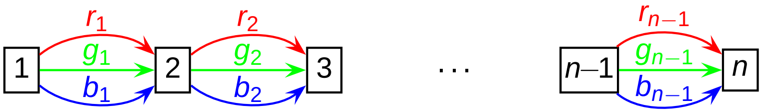
Generalizing from Chains



- In a chain a *forward edge* e_{xy} for $y > x + 1$ might replace some minimal paths but doesn't contribute any additional minimal path weights to worst case (as assuming each edge has a single colour).
- Can ignore edges e_{x1} to source and edges e_{ny} from destination (as assuming each edge has a positive weight).
- Any *backward edge* e_{xy} for $y < x$ appears to really complicate situation, get many more paths, so probably many more minimal paths...

or are there?

Generalizing from Chains

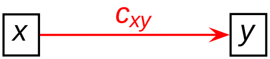


- In a chain a *forward edge* e_{xy} for $y > x + 1$ might replace some minimal paths but doesn't contribute any additional minimal path weights to worst case (as assuming each edge has a single colour).
- Can ignore edges e_{x1} to source and edges e_{ny} from destination (as assuming each edge has a positive weight).
- Any *backward edge* e_{xy} for $y < x$ appears to really complicate situation, get many more paths, so probably many more minimal paths...
  or are there?

Canonical Weighted Coloured-Edge Graphs

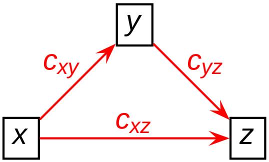
Call weighted coloured-edge graph $\mathbf{G} = \langle V, E, \omega, \lambda\rangle$ *canonical* if:

- **G** is *complete* in each colour:
  for all vertices $x \neq y$ and colour c there is exactly one edge c_{xy} from x to y with $\lambda(c_{xy}) = c$.



- **G** satisfies the *triangle inequality* in each colour:
  for all distinct vertices x, y, z and colour c, the triangle formed by the three edges c_{xy}, c_{yz}, c_{xz} with $\lambda(c_{xy}) = \lambda(c_{yz}) = \lambda(c_{xz}) = c$ obeys

$$\omega(c_{xz}) \leq \omega(c_{xy}) + \omega(c_{yz}).$$



Canonization of Weighted Coloured-Edge Graphs

Lemma: Canonical Form

Given any weighted coloured-edge graph $\mathbf{G} = \langle V, E, \omega, \lambda\rangle$ there is a canonical graph $\mathbf{G}^*$ with the same vertices and colours that has at least as many minimal path weights.

Justification:

- Can complete **G** by adding edges c_{xy} with weight $n \cdot w$ where w is the maximum weight of any edge in V, added edges won't affect any existing minimal paths.
- Can iteratively reduce weights of edges to obey triangle inequality if $\omega(c_{xz}) > \omega(c_{xy}) + \omega(c_{yz})$ by taking $\omega(c_{xz}) = \min\{\omega(c_{xy}) + \omega(c_{yz}) \mid y \in V\}$, process eventually terminates in finite graph.

AUT UNIVERSITY
The University for the changing world.

Upper bound for Weighted Coloured-Edge Graphs

Theorem: Upper bound on number of minimal path weights
A weighted coloured-edge graph **G** with n vertices and k colours can have at most k^{n-1} minimal path weights.

Notes:

- Same bound as for chains - what happened to the complexity introduced by back edges?
- Might have a factorial number of paths, but very few of them can give minimal paths.
- Alterative paths might have the same minimal path weights (might be important if application wants to use criteria based not only on path weights).
- Proof very dependent on fact that individual edges have single colour (so won't work for multi-weighted graphs).

Proof Outline

- Replace $\mathbf{G}$ with its canonical form $\mathbf{G}^*$, by the lemma $\mathbf{G}^*$ has at least as many minimal path weights.
- For each colour c count the number of minimal path weights $f_c(n)$ from u to v in $\mathbf{G}^*$ that start at u with an edge c_{ux} for which $\lambda(c_{ux}) = c$.
- Observation: if there is a minimal path starting with c_{ux} that passes through vertex y and if $\lambda(c_{ux}) = \lambda(c_{uy})$ then $\omega(c_{ux}) < \omega(c_{uy})$.
- Can assume no minimal weight path has two consecutive edges with same colour (by triangle inequality).
- Use induction to show that $f_c(n) \leq k^{n-2}$ for each of the k colours c. Clearly $f_c(2) = 1$ for every canonical graph with only two vertices.

AUT UNIVERSITY
The University for the changing world.

Inductive step of proof to show $f_c(n) \leq k^{n-2}$

Inductive step in canonical graph with $n+1$ vertices:

- Order the intermediate vertices $v_1, v_2, \ldots, v_{n-1}$ so that $\omega(c_{ux}) \leq \omega(c_{uy})$ if $x < y$. Can assume no minimal weight path starting with c_{ux} passes through vertex y for $y < x$ (by earlier observation).
- Minimal weight path starting with e_{u1} changes colour for next edge, by induction there are at most $\Sigma_{c' \neq c} f_{c'}(n) \leq (k-1)k^{n-2}$ such minimal path weights.
- Minimal weight path starting with e_{u2} changes colour for next edge, by induction there are at most $\Sigma_{c' \neq c} f_{c'}(n-1) \leq (k-1)k^{n-3}$ such minimal path weights.
- $\ldots$
- Total $f_c(n+1) \leq (k-1)k^{n-2} + (k-1)k^{n-3} + \cdots + 1 = k^{n-1}$.

Summary

- Weighted Coloured-Edge Graphs.
- Applications where objective expressible as a k-ary increasing function of total weights in each mode.
- Chains can potentially give k^{n-1} minimal paths, not so bad in practice.
- Chains are as bad as it gets for weighted coloured-edge graphs.

- Current work on average-case analysis

AUT UNIVERSITY
The University for the changing world.