Scheduling of Virtual Connections in Fast Networks

Thomas Erlebach
Technische Universität München
erlebach@informatik.tu-muenchen.de

Klaus Jansen
Universität Trier
jansen@dm5.uni-trier.de

- Unit Duration and Unit Bandwidth
  - Bidirectional Calls in Trees and Rings
  - Directed Calls in Trees and Rings
- Arbitrary Duration and Arbitrary Bandwidth

Motivation

- ATM Networks (ATM $=$ *Asynchronous Transfer Mode*)
  - high bandwidth
  - guaranteed quality of service
  - bandwidth reservation schemes
  - *virtual* connections (calls) share physical links
- ATM can improve communication performance in workstation clusters and make them suitable for parallel computation.
- Problem: Given a set of connection requests, how do we schedule these calls in order to minimize the completion time?

An instance of the call-scheduling problem is given by:

(a) a graph $G = (V, E)$ with edge capacities $c : E \longrightarrow \mathbb{N}$

(unit bandwidth: $\forall e \in E : c(e) = 1$)

- undirected graph (bidirectional calls, bidirectional links)
- symmetric directed graph (directed calls, bidirectional links)
- directed graph (directed calls, directed links)

(b) a set R of call requests $r = (u, v, b, d)$, where

- u is the sender
- v is the receiver
- b is the requested bandwidth (unit bandwidth: $b = 1$)
- d is the duration of the call (unit duration: $d = 1$)

A feasible schedule assigns to each request $r = (u, v, b, d)$:

(1) a (directed) path P_r from u to v in G

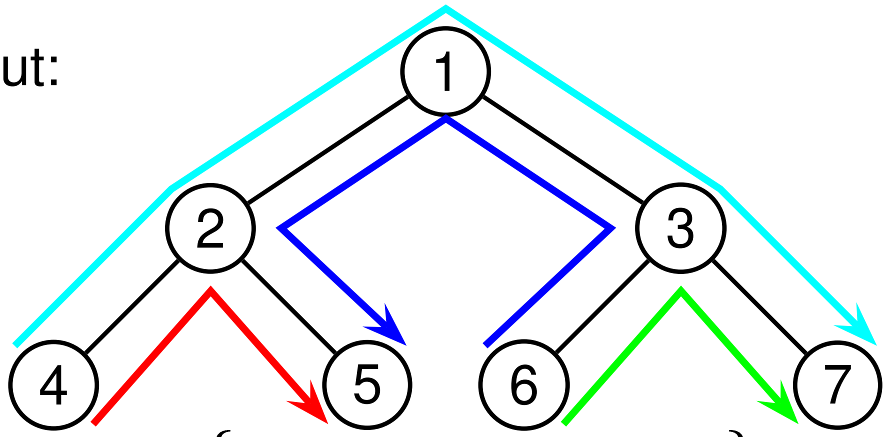
(2) a starting time t_r (preemption is not allowed!)

satisfying:

The sum of bandwidths of simultaneously active calls using an edge e must not exceed $c(e)$.

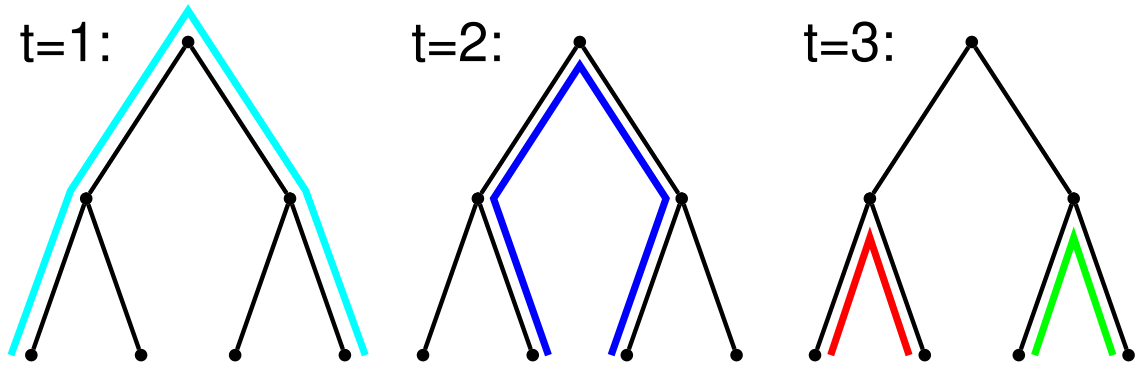
Example $(b = 1, d = 1)$

Input:

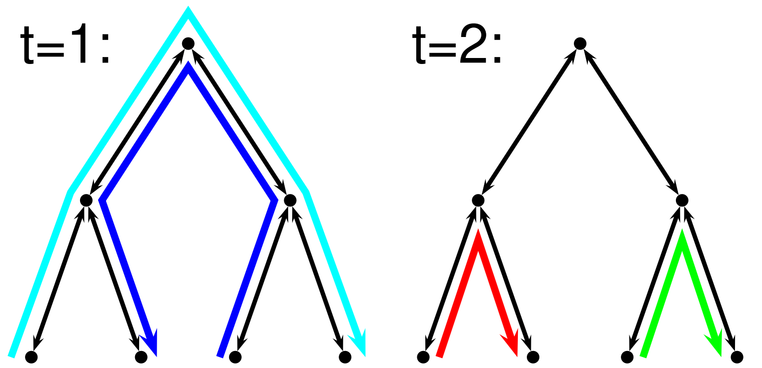


R={(4,7),(4,5),(6,7),(6,5)}

Schedule for Bidirectional Calls:



Schedule for Directed Calls:



Scheduling Bidirectional Calls in Binary Trees

Definition: Let $G = (V, E)$ be a tree and R a set of call requests. For $v \in V$ we denote by R_v the subset of R which contains all calls that touch node v:

$$R_v := \{r \in R | v \in \text{PATH}(r)\}$$

Algorithm

- select an arbitrary node as the root node
- compute the schedule time step by time step:
  - for each time step, process the nodes in depth-first order
  - when processing node v, schedule up to two calls in R_v, giving priority to calls using both downward links from v

Theorem: This algorithm computes a schedule of minimum length.

Scheduling Bidirectional Calls in Arbitrary Trees

Theorem: Given schedules S_v of length l_v for each individual set R_v, these schedules can be merged into a schedule for R in polynomial time. The length of the resulting schedule is $\max\{l_v | v \in V\}$.

Sketch of proof:

- visit the nodes in depth-first order
- at node v, permute S_v to make it compatible with the existing schedule and take the union of the two.

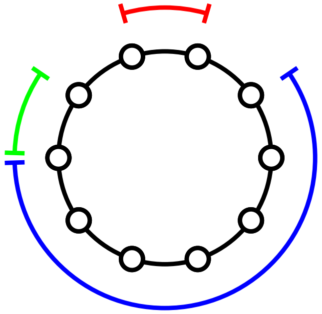
Scheduling Bidirectional Calls in Arbitrary Trees (2)

Theorem: [Golumbic/Jamison 1985] Scheduling calls in a star-like graph is equivalent to edge-coloring a multigraph. As a consequence, the call-scheduling problem for arbitrary trees is $\mathcal{NP}$–hard.

Theorem: There is a polynomial time 1.1–approximation algorithm for call-scheduling in arbitrary trees.

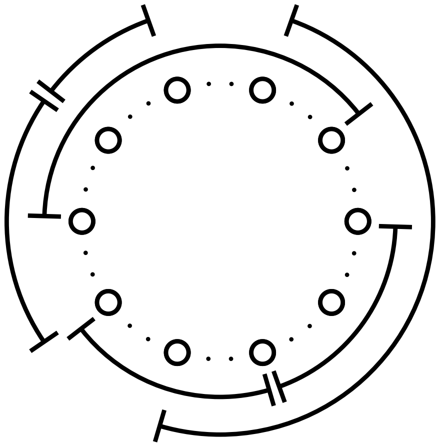
Sketch of proof: Use the 1.1–approximation algorithm for edge-coloring from [Nishizeki/Kashiwagi 1990] to schedule the calls in each set R_v, and combine these individual schedules with the merging algorithm.

Scheduling Bidirectional Calls in Ring Networks



- Each call can be routed clockwise or counterclockwise.
- If the routing is determined by ignoring some arbitrary link of the ring, the resulting schedule is a 2–approximation.

Coloring circular arcs: a related problem?



ARC-COLORING: Given a family F of circular arcs and a positive integer k, can F be colored with k colors?

Theorem: ARC-COLORING is $\mathcal{NP}$–complete.
(in [Garey, Johnson, Miller, Papadimitriou 1980])

Theorem: Scheduling bidirectional calls in ring networks optimally is $\mathcal{NP}$–hard.

Proof: by reduction from ARC-COLORING

Let a family F of circular arcs and a positive integer k be given.

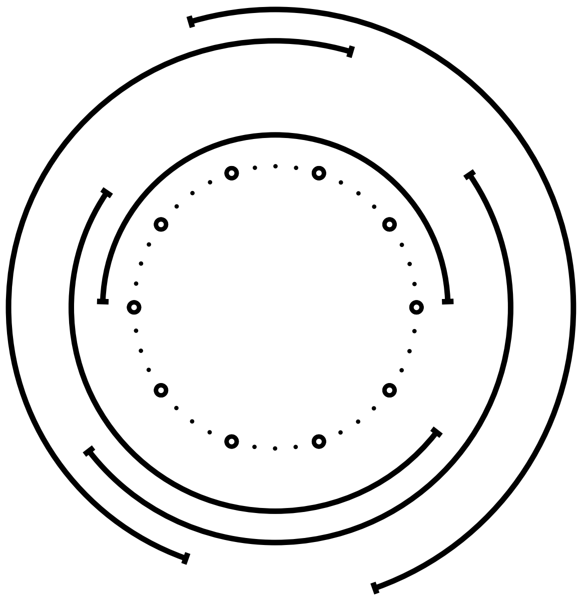
(1) extend F so that every part of the circle is spanned by exactly k arcs

(2) select two intervals of the circle lying opposite each other, and split each arc overlapping one of these intervals into three separate arcs which must be assigned the same color in any k–coloring of F

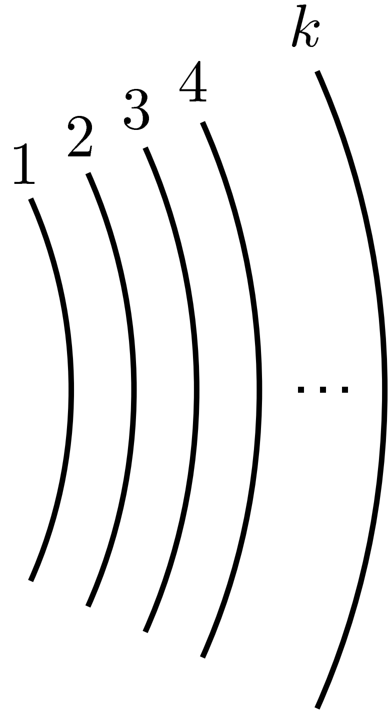
(3) interpret each arc as a call between its endpoints

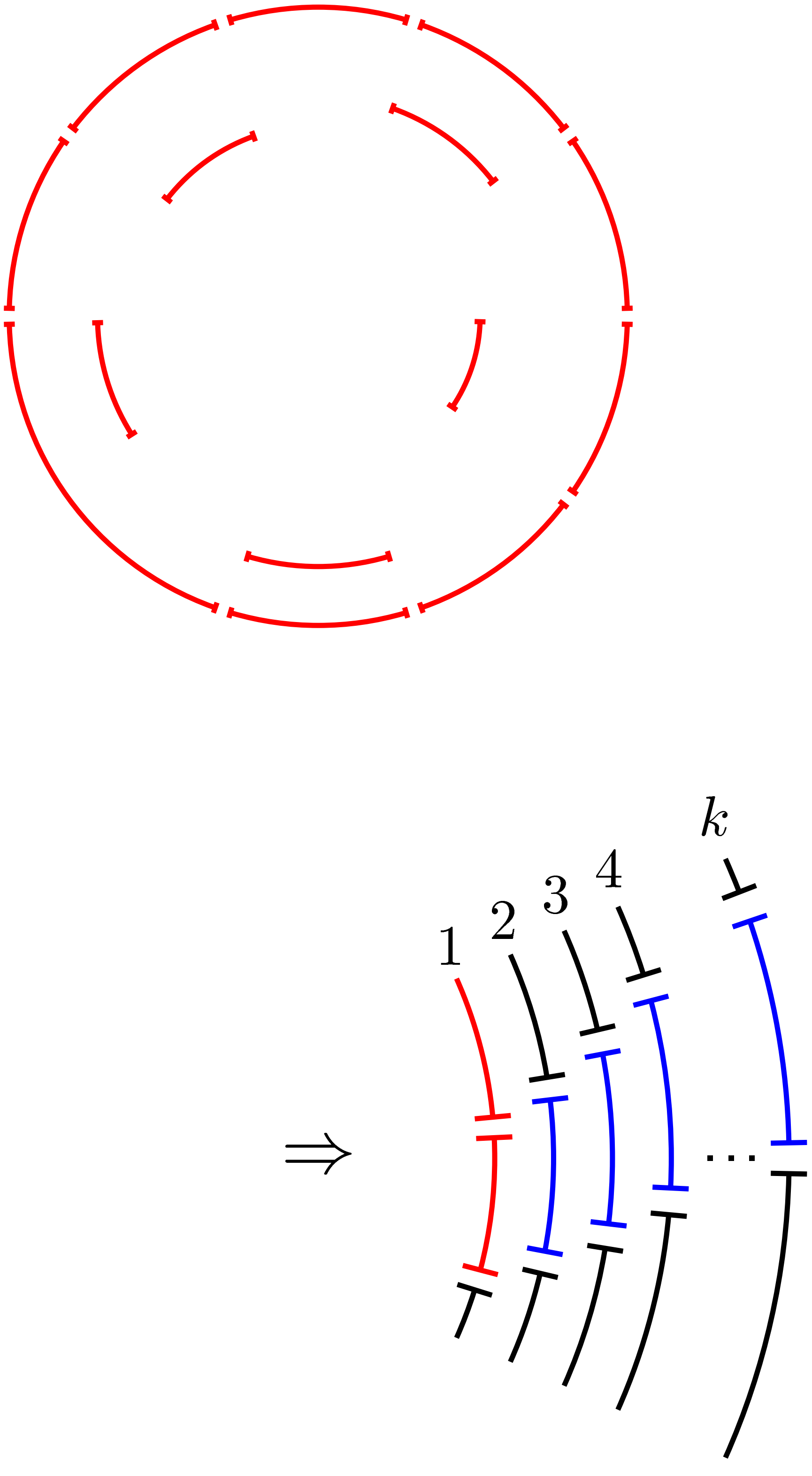
Step (1):

$k = 4$



Step (2):





1
2
3
4
k

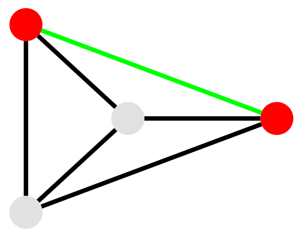
Scheduling Directed Calls in Tree Networks

Theorem: [Mihail, Kaklamanis, Rao 1995] There is a polynomial time $\frac{15}{8}$–approximation algorithm for scheduling directed calls in tree networks.

Theorem: The scheduling problem for directed calls in tree networks is $\mathcal{NP}$–hard.

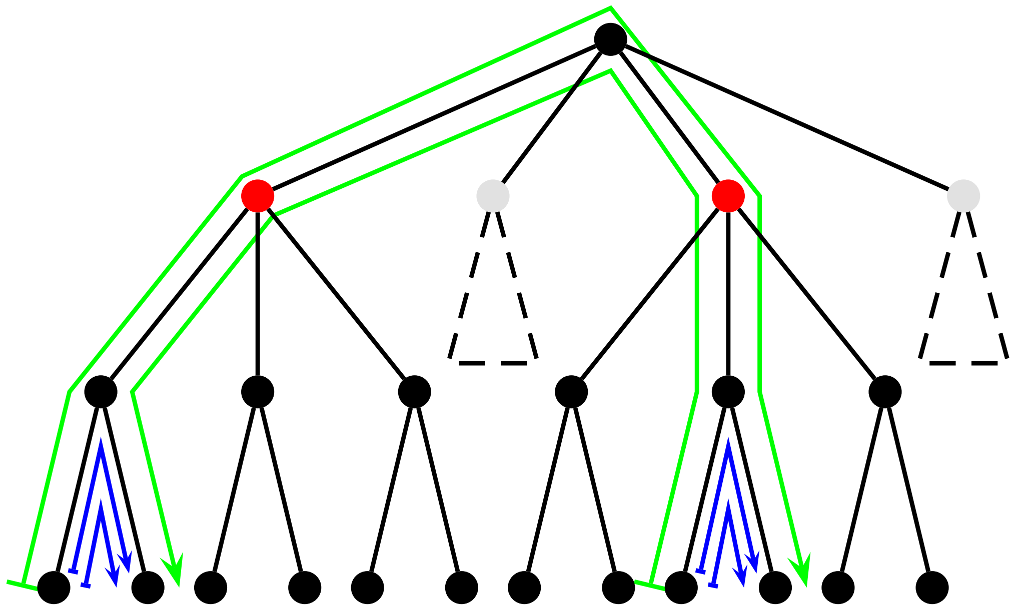
Proof: by reduction from EDGE-COLORING

3–regular graph



chromatic index:
3 or 4 ?

$\Rightarrow$



minimum schedule length: 3 or 4 ?

Scheduling Directed Calls in Ring Networks

Optimization Problem:

- can be shown $\mathcal{NP}$–hard by a reduction from ARC-COLORING (similar to the bidirectional case, with each arc being viewed as two directed calls in opposite direction between its endpoints)

Approximation Algorithm:

- ignore some arbitrary link of the ring $\Rightarrow$ 2–approximation

Summary: Unit Bandwidth and Unit Duration

	bidirectional	directed
linear arrays	polynomial	polynomial
binary trees	polynomial	$\mathcal{NP}$–hard $\frac{7}{4}$–approximation
trees	$\mathcal{NP}$–hard 1.1–approximation	$\mathcal{NP}$–hard $\frac{15}{8}$–approximation
rings	$\mathcal{NP}$–hard 2–approximation	$\mathcal{NP}$–hard 2–approximation

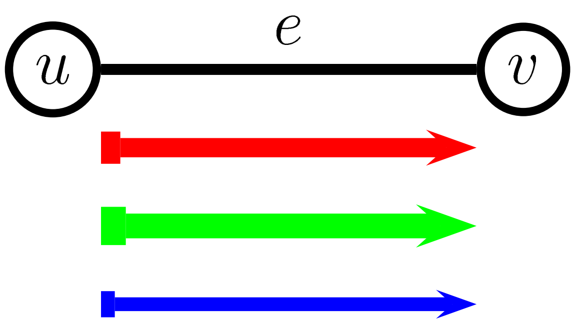
Arbitrary Bandwidth and Arbitrary Duration

The **PARTITION** Problem

Given a set A and sizes $s(a)$ for all $a \in A$ such that $\sum s(a) = 2 \cdot B$, is there a subset $A' \subset A$ satisfying $\sum_{a \in A'} s(a) = B$?

Theorem: Scheduling calls with unit duration and arbitrary bandwidth optimally is already $\mathcal{NP}$–hard on a single link.

Proof: by reduction from PARTITION



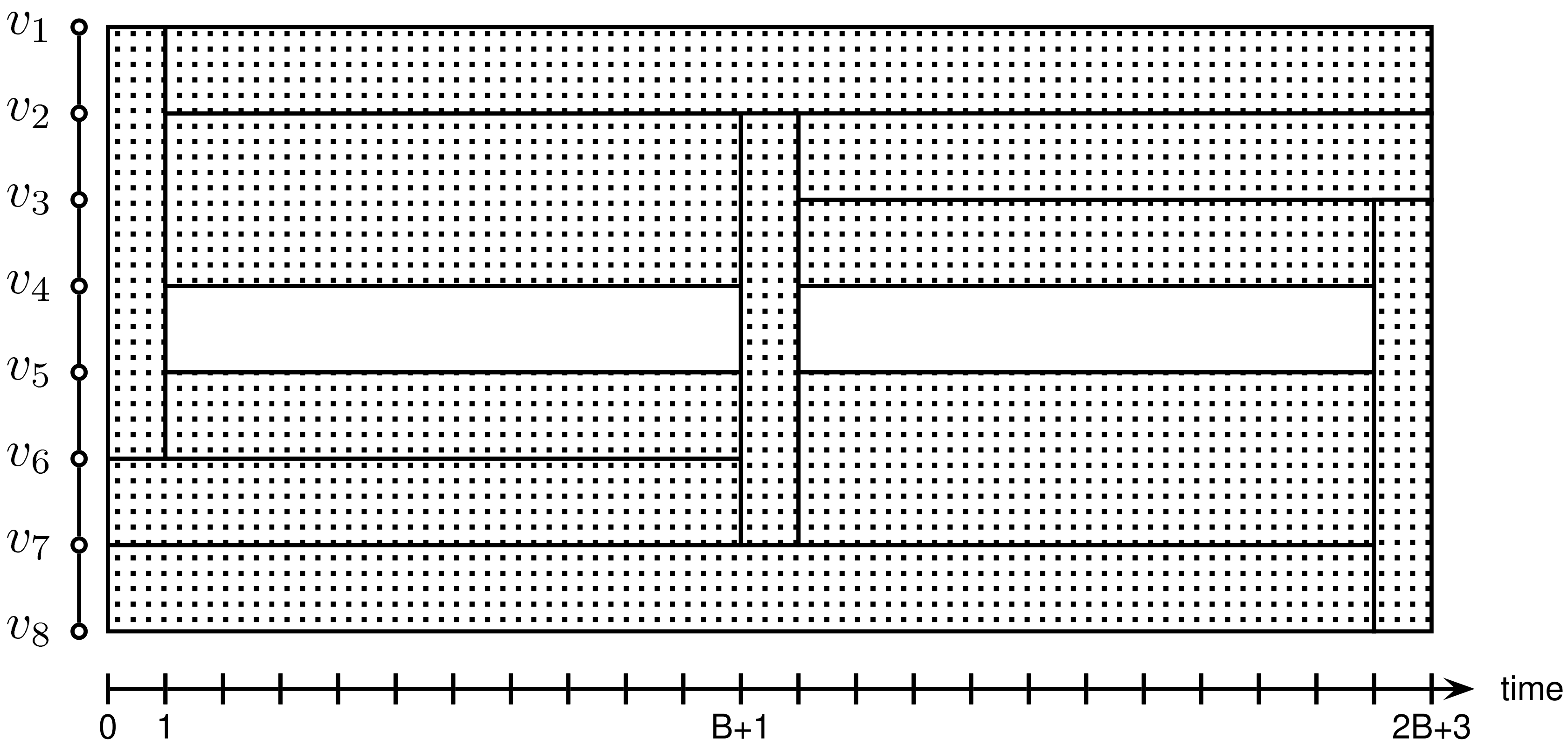
$$c(e) = B$$

$$a \in A \quad \mapsto \quad (u, v, s(a), 1) \in R$$

minimum schedule length: 2 or 3 ?

Theorem: Scheduling calls with arbitrary duration and unit bandwidth optimally is $\mathcal{NP}$–hard on a linear array with 8 nodes.

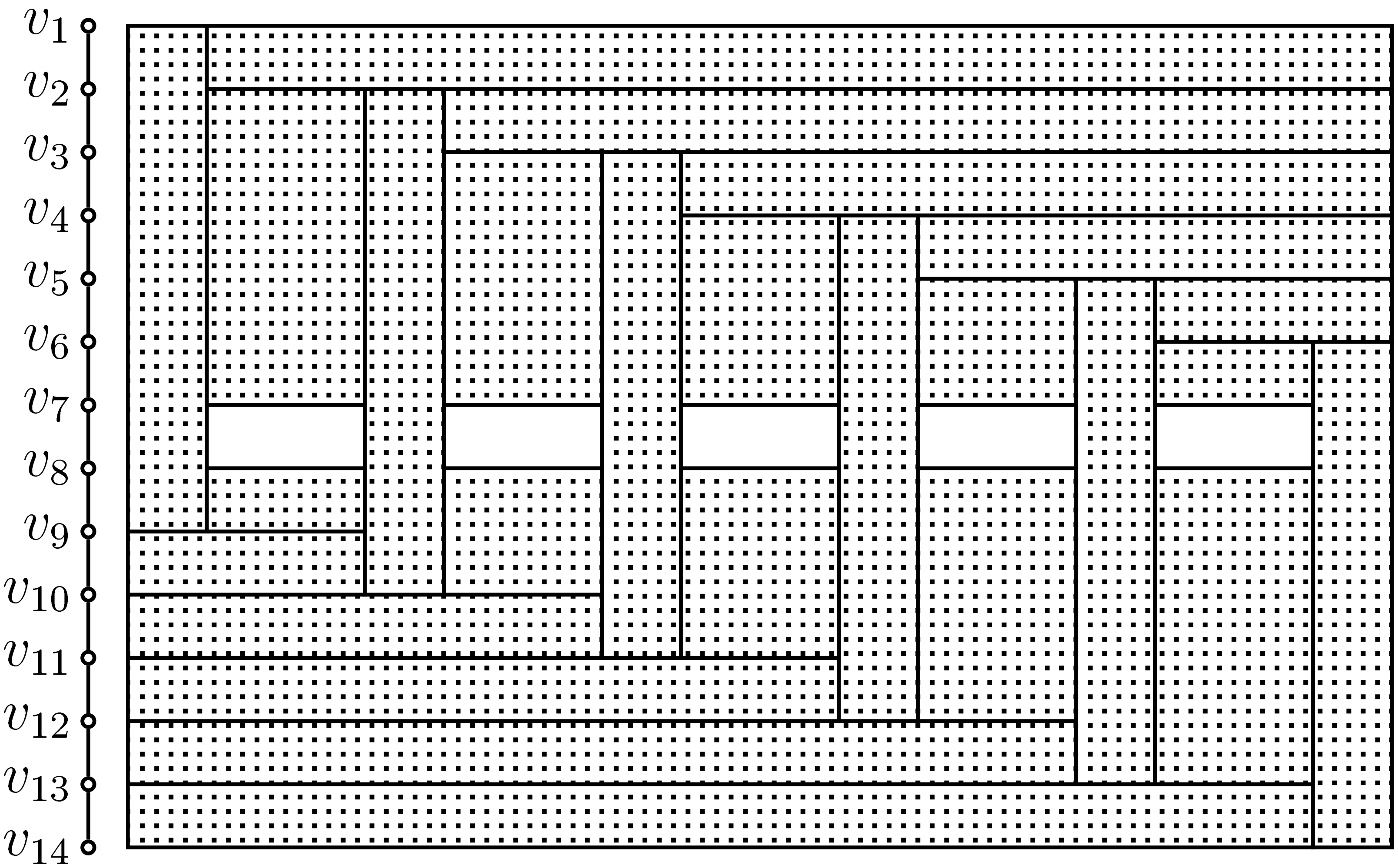
Proof: reduction from PARTITION, $a \in A \mapsto (v_4, v_5, 1, s(a)) \in R$



minimum schedule length $\leq 2B + 3$?

Reduction from 3-PARTITION $\Rightarrow \mathcal{NP}$–complete in the strong sense

(as in [Doenhardt/Lengauer 1987] for 1D layout compaction problem)



Let $G = (V, E)$ be a network with $k \geq 2$ edge-disjoint paths between some node u and some node v.

Theorem: Optimal scheduling of calls with arbitrary duration and unit bandwidth is $\mathcal{NP}$–hard in G.

Proof: by reduction from PARTITION

- $a \in A \mapsto (u, v, 1, s(a)) \in R$
- $k - 2$ additional calls $(u, v, 1, B) \in R$

minimum schedule length $\leq B$?

Theorem: Optimal scheduling of calls with unit duration and arbitrary bandwidth is $\mathcal{NP}$–hard in G.

Proof: by reduction from PARTITION (similar to above)

Conclusion and Outlook

Results for the setting with unit duration and unit bandwidth are applicable for the wavelength routing problem in optical networks.

Trying to obtain call-scheduling algorithms that are useful in practice, we face the following challenges:

- more network topologies, e.g. meshes
- algorithms for arbitrary bandwidth and arbitrary duration
- on-line problems without knowledge of call durations and future call requests (evaluation by competitive analysis)