Construction of yearly schedules for train drivers

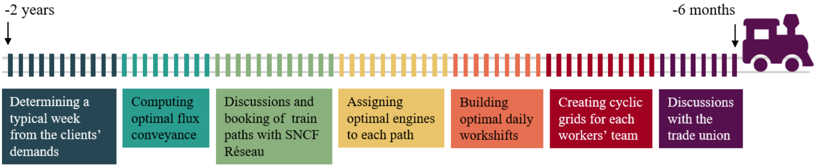
Héloïse Gachet
Supervisor: Frédéric Meunier

Plan

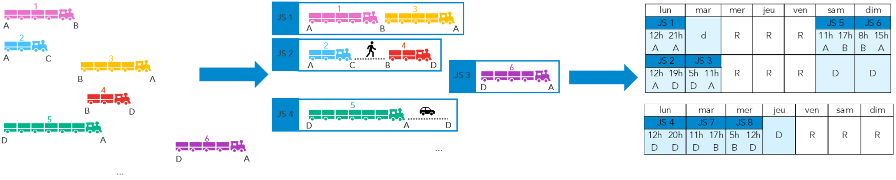
Introduction

- 18% of rail freight in Europe (10% in France)
- Between 1800 and 2000 trains per week
- Many differences with passengers transportation...
  - Trains mostly at night
  - Priority goes to passengers
  - Client satisfaction: delivery on time
  - Key goal: reducing operational costs

Resource planning



Problem description



High combinatorial complexity

National input:

1 800 trains

1 000 repositionings $\rightarrow$ 200 000 workshifts $\rightarrow$ ... "shift blocks"

Linear Programming

The "best" cocktail ever
a.k.a. continuous knapsack problem

1-Orange juice
40 mL
★★★

2-Grenadine
10 mL
★★★★

3-Coconut milk
60 mL
★★★★★

4-Lemonade
20 mL
★★

$$\text{Max} \quad \sum_{i\in[4]} \star_i \, x_i$$

$$\text{s.c.} \quad \sum_{i\in[4]} x_i \leqslant V_{tot}$$

$$x_i \in [0, v_i] \quad \forall i \in [4]$$

$$\downarrow$$

Mixed Integer Linear Programming

The "best" tea-time ever

a.k.a. 0-1 knapsack problem

1-Best cocktail
1kg
😄 😄 😄 😄 😄

2-Cookies
3kg
😄 😄 😄 😄

3-Board game
1.5kg
😄 😄 😄

4-Chocolate
0.5kg
😄 😄

$$\text{Max} \quad \sum_{i\in[4]} \text{😄}_i \, x_i$$

$$\text{s.c.} \quad \sum_{i\in[4]} w_i x_i \leqslant W_{tot}$$

$$x_i \in \{0,1\} \quad \forall i \in [4]$$

$$\downarrow$$

Modeling and numerous variables

Bin-packing problem

$$\begin{aligned}
\text{Min} \quad & \sum_i y_j \\
\text{s.c.} \quad & \sum_j x_{ij} = 1 \quad \forall i \\
& \sum_i w_i x_{ij} \leqslant W_{tot} y_j \quad \forall j \\
& x_{ij} \in \{0,1\} \quad \forall i,j \\
& y_j \in \{0,1\} \quad \forall j
\end{aligned}$$

OR

$$\begin{aligned}
\text{Min} \quad & \sum_c z_c \\
\text{s.c.} \quad & \sum_c n_{i \in c} z_c = N_{tot,i} \quad \forall i \\
& z_c \in \mathbb{N} \quad \forall c \in \mathcal{C}
\end{aligned}$$

Model

$$
\begin{aligned}
\text{Min} \quad & \sum_{w \in W} \sum_{b \in B_w} c_b x_{b,w} \\
\text{s.c.} \quad & \sum_{w \in W} \sum_{\substack{b \in B_w \\ t \in b}} x_{b,w} \geqslant 1 \quad && \forall t \in T \\
& x_{b,w} \in \{0,1\} && \forall w \in W, \forall b \in B_w
\end{aligned}
$$

B_w = set of feasible "shift blocks" for the workers' team w
→ combinatorial explosion

Column Generation: standard resolution methodology of a linear program when the number of variables is large

Variables Generation

Generation of variables throughout the resolution

Issue: Solving the pricing sub-problem quickly

For our problem:

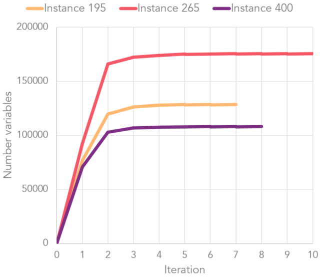
Generation of a new variable = Solving a Resource Constrained Shortest Path problem

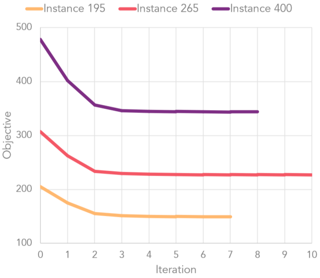
→ NP-hard problem

→ Modeling challenges

Results

	Objective			Total time	Nb iterations	Nb variables
	Sequential	Lower bound	Upper bound			
Instance 195 trains	176	149.9	**151 (-14%)**	15min31	8	128 527
Instance 265 trains	261	227.3	**229 (-12%)**	28min13	11	175 716
Instance 400 trains	402	344.2	**347 (-13%)**	32min46	9	108 150





Plan

Cyclic grids

lun	mar	mer	jeu	ven	sam	dim
JS 1 12h 21h A A	d	R	R	R	JS 5 11h 17h A B	JS 6 8h 15h B A
JS 2 12h 19h A D	JS 3 5h 11h D A	R	R	R	D	D

lun	mar	mer	jeu	ven	sam	dim
JS 4 12h 20h D D	JS 7 11h 17h D B	JS 8 5h 12h B D	D	R	R	R

Introduction

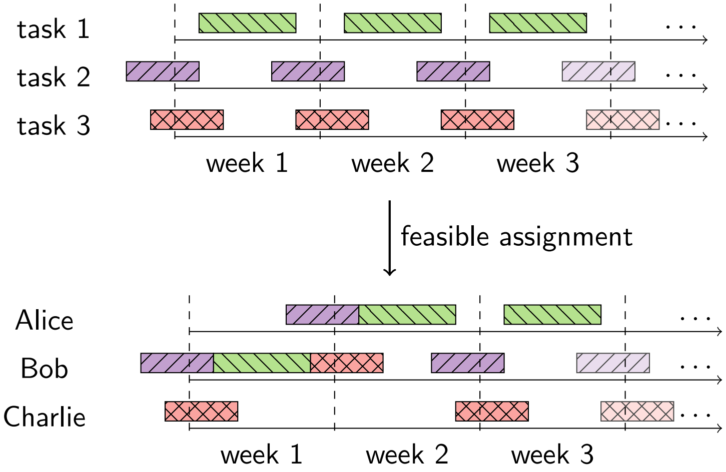
Input:

- Tasks to be performed at the same time every week
- Indistinguishable workers

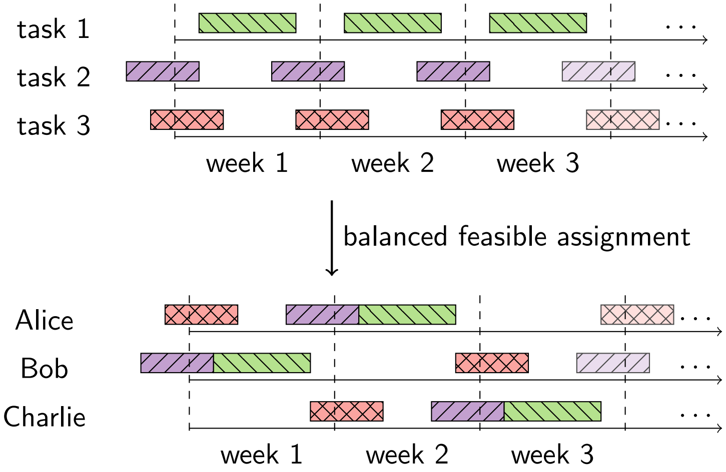
Output: Assignment of tasks to workers so that each worker performs each task with same asymptotic frequency

e.g., Monday from 9:10 to 10:30, drive train 9015 from Paris to Lyon with Alice, Bob, or Charlie

Feasible assignment



Balanced assignment: an example



Balanced assignment: formally

Input: n tasks to be repeated every week, q workers

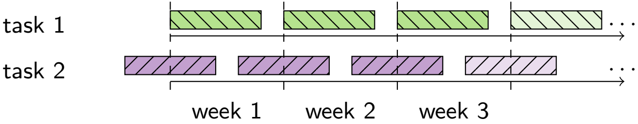
Assignment: $f\colon [n] \times \mathbb{Z}_{>0} \to [q]$, where $f(i,r) = j$ means worker j performs task i on week r

Feasible assignment: f is non overlapping

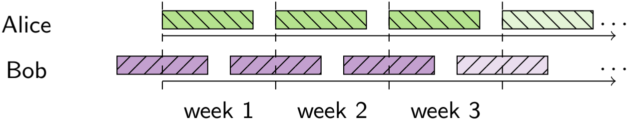
Balanced assignment:

$$\lim_{t \to +\infty} \frac{1}{t} \left| \{ r \in [t] \colon f(i,r) = j \} \right| = \frac{1}{q} \qquad \forall i, j$$

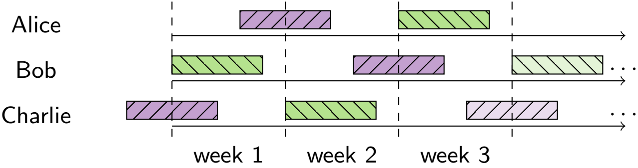
Balanced assignments: unachievable case



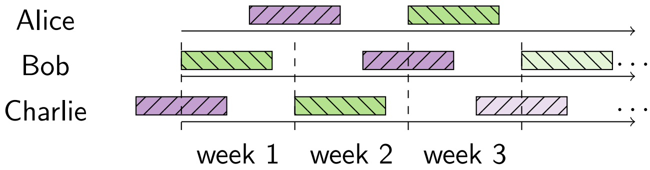
With 2 workers



With 3 workers



First results



Theorem G.–Meunier 2024+

There exists a balanced feasible assignment if and only if there exists a feasible assignment with a worker performing each task at least once.

$\rightarrow$ *Elementary proof*

Side note on complexity

- Polynomial problems
  ex: The "best" cocktail i.e. continuous knapsack problem
- NP-hard problems
  - Weakly
    ex: The "best" tea-time i.e. 0-1 knapsack problem
  - Strongly
    ex: Bin-packing problem

One of the Millennium Prize Problems: P $=$ NP?

Side note on complexity

- Decision problem
  ex: Given $n \in \mathbb{N}$, does it have a non-trivial factor?
  $\rightarrow$ polynomial [1]
- Construction problem
  ex: Given $n \in \mathbb{N}$, find a non-trivial factor of n if it exists?
  $\rightarrow$?

[1] https://en.wikipedia.org/wiki/AKS_primality_test

First results

Theorem G.–Meunier 2024+

Deciding whether there exists a balanced feasible assignment can be done in polynomial time.

Theorem G.–Meunier 2024+

Suppose the number of workers fixed. Building a balanced feasible assignment, if it exists, can be done in polynomial time.

Richer extension

- Maximal number of tasks per week

Theorem G.–Meunier 2024+

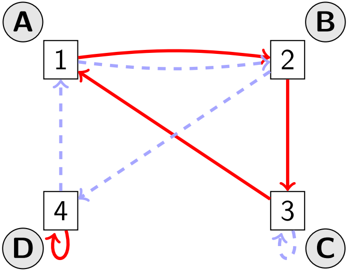
If there exists a feasible assignment with a worker performing each task at least once and all workers are busy at time zero, then there exists a balanced feasible assignment.

Moving pebbles

Consider:

- graph $D = (V, A)$ with colors on the arcs, each color on a collection of vertex disjoint cycles covering V
- one pebble on each vertex

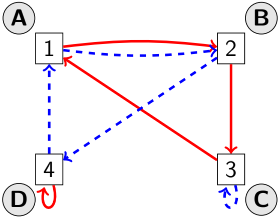
Sequence of colors defines sequence of moves of the pebbles



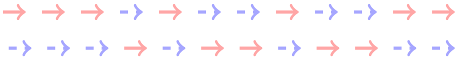
Color: →

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

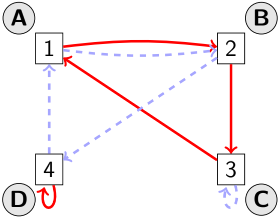


arcs visiting:

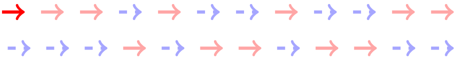
	1 → 2	2 → 3	3 → 1	4 → 4	1 ⇢ 2	2 ⇢ 4	3 ⇢ 3	4 ⇢ 1
A	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

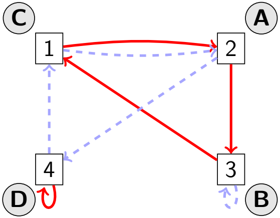


arcs visiting:

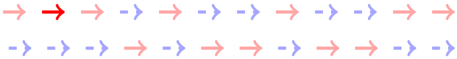
	1 →2	2 →3	3 →1	4 →4	1 -›2	2 -›4	3 -›3	4 -›1
A	**1**	0	0	0	0	0	0	0
B	0	**1**	0	0	0	0	0	0
C	0	0	**1**	0	0	0	0	0
D	0	0	0	**1**	0	0	0	0

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

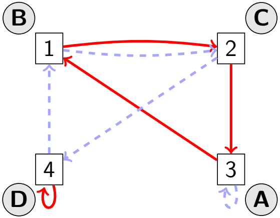


arcs visiting:

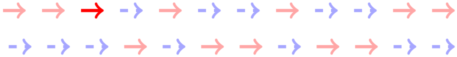
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	**1**	0	0	0	0	0	0
B	0	1	**1**	0	0	0	0	0
C	**1**	0	1	0	0	0	0	0
D	0	0	0	**2**	0	0	0	0

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

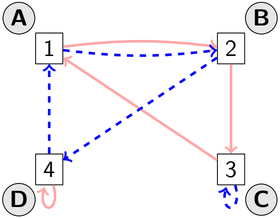


arcs visiting:

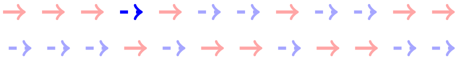
	1 → 2	2 → 3	3 → 1	4 → 4	1 ⇢ 2	2 ⇢ 4	3 ⇢ 3	4 ⇢ 1
A	1	1	**1**	0	0	0	0	0
B	**1**	1	1	0	0	0	0	0
C	1	**1**	1	0	0	0	0	0
D	0	0	0	**3**	0	0	0	0

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

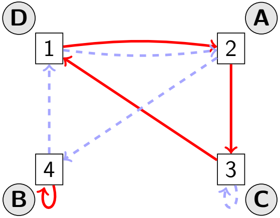


arcs visiting:

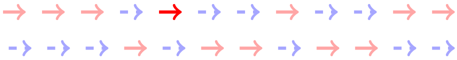
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	1	1	0	**1**	0	0	0
B	1	1	1	0	0	**1**	0	0
C	1	1	1	0	0	0	**1**	0
D	0	0	0	3	0	0	0	**1**

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

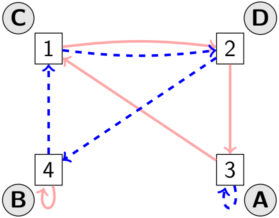


arcs visiting:

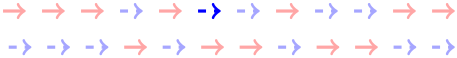
	1→2	2→3	3→1	4→4	1-→2	2-→4	3-→3	4-→1
A	1	**2**	1	0	1	0	0	0
B	1	1	1	**1**	0	1	0	0
C	1	1	**2**	0	0	0	1	0
D	**1**	0	0	3	0	0	0	1

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

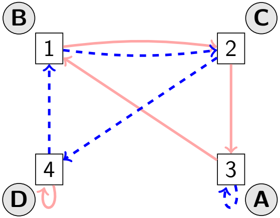


arcs visiting:

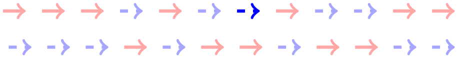
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	1	0	1	0	**1**	0
B	1	1	1	1	0	1	0	**1**
C	1	1	2	0	**1**	0	1	0
D	1	0	0	3	0	**1**	0	1

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

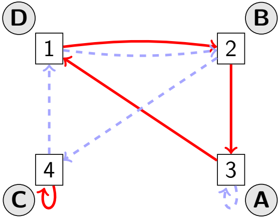


arcs visiting:

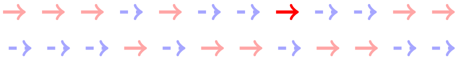
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	1	0	1	0	**2**	0
B	1	1	1	1	**1**	1	0	1
C	1	1	2	0	1	**1**	1	0
D	1	0	0	3	0	1	0	**2**

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

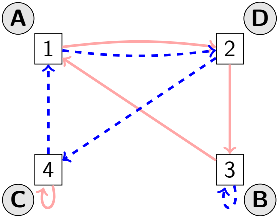


arcs visiting:

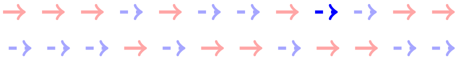
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	**2**	0	1	0	2	0
B	1	**2**	1	1	1	1	0	1
C	1	1	2	**1**	1	1	1	0
D	**2**	0	0	3	0	1	0	2

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

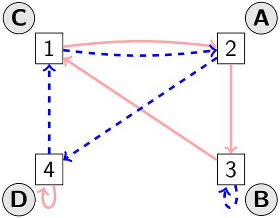


arcs visiting:

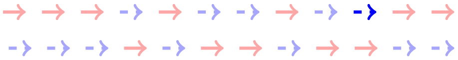
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	2	0	**2**	0	2	0
B	1	2	1	1	1	1	**1**	1
C	1	1	2	1	1	1	1	**1**
D	2	0	0	3	0	**2**	0	2

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

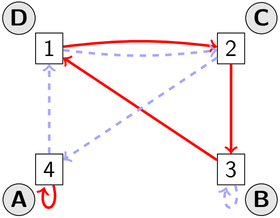


arcs visiting:

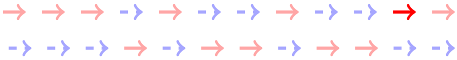
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	2	0	2	**1**	2	0
B	1	2	1	1	1	1	**2**	1
C	1	1	2	1	**2**	1	1	1
D	2	0	0	3	0	2	0	**3**

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

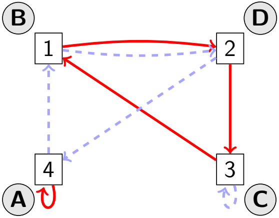


arcs visiting:

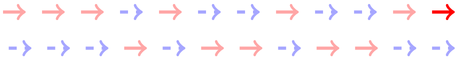
	1 →2	2 →3	3 →1	4 →4	1 -›2	2 -›4	3 -›3	4 -›1
A	1	2	2	**1**	2	1	2	0
B	1	2	**2**	1	1	1	2	1
C	1	**2**	2	1	2	1	1	1
D	**3**	0	0	3	0	2	0	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

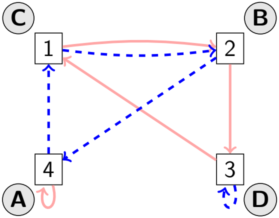


arcs visiting:

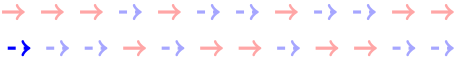
	1 → 2	2 → 3	3 → 1	4 → 4	1 -› 2	2 -› 4	3 -› 3	4 -› 1
A	1	2	2	**2**	2	1	2	0
B	**2**	2	2	1	1	1	2	1
C	1	2	**3**	1	2	1	1	1
D	3	**1**	0	3	0	2	0	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

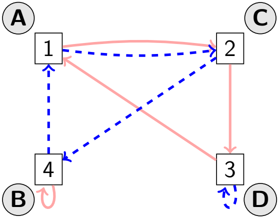


arcs visiting:

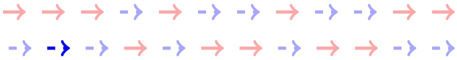
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	2	2	2	1	2	**1**
B	2	2	2	1	1	**2**	2	1
C	1	2	3	1	**3**	1	1	1
D	3	1	0	3	0	2	**1**	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

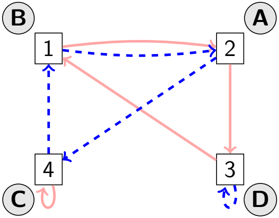


arcs visiting:

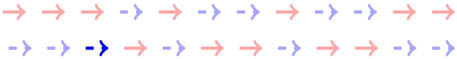
	1→2	2→3	3→1	4→4	1-→2	2-→4	3-→3	4-→1
A	1	2	2	2	**3**	1	2	1
B	2	2	2	1	1	2	2	**2**
C	1	2	3	1	3	**2**	1	1
D	3	1	0	3	0	2	**2**	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

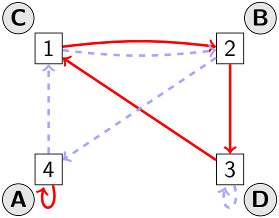


arcs visiting:

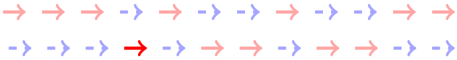
	1 → 2	2 → 3	3 → 1	4 → 4	1 ⇢ 2	2 ⇢ 4	3 ⇢ 3	4 ⇢ 1
A	1	2	2	2	3	**2**	2	1
B	2	2	2	1	**2**	2	2	2
C	1	2	3	1	3	2	1	**2**
D	3	1	0	3	0	2	**3**	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

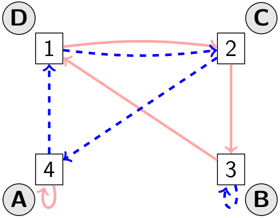


arcs visiting:

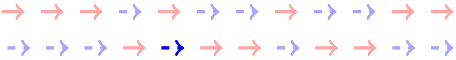
	1 →2	2 →3	3 →1	4 →4	1 -›2	2 -›4	3 -›3	4 -›1
A	1	2	2	**3**	3	2	2	1
B	2	**3**	2	1	2	2	2	2
C	**2**	2	3	1	3	2	1	2
D	3	1	**1**	3	0	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

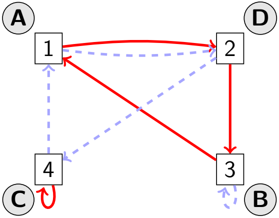


arcs visiting:

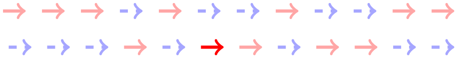
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	1	2	2	3	3	2	2	**2**
B	2	3	2	1	2	2	**3**	2
C	2	2	3	1	3	**3**	1	2
D	3	1	1	3	**1**	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

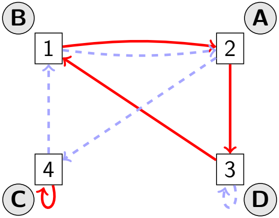


arcs visiting:

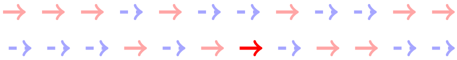
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	**2**	2	2	3	3	2	2	2
B	2	3	**3**	1	2	2	3	2
C	2	2	3	**2**	3	3	1	2
D	3	**2**	1	3	1	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

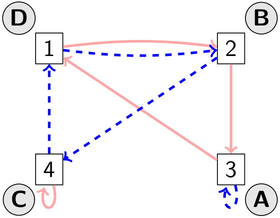


arcs visiting:

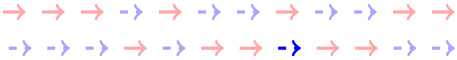
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	2	**3**	2	3	3	2	2	2
B	**3**	3	3	1	2	2	3	2
C	2	2	3	**3**	3	3	1	2
D	3	2	**2**	3	1	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

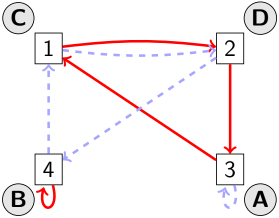


arcs visiting:

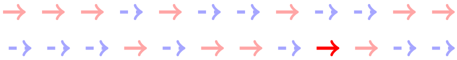
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	2	3	2	3	3	2	**3**	2
B	3	3	3	1	2	**3**	3	2
C	2	2	3	3	3	3	1	**3**
D	3	2	2	3	**2**	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

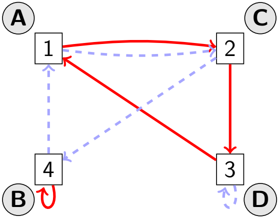


arcs visiting:

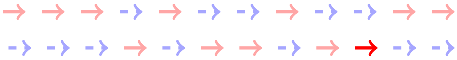
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	2	3	**3**	3	3	2	3	2
B	3	3	3	**2**	2	3	3	2
C	**3**	2	3	3	3	3	1	3
D	3	**3**	2	3	2	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

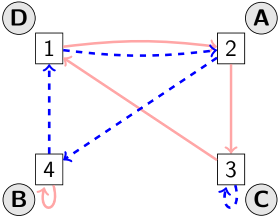


arcs visiting:

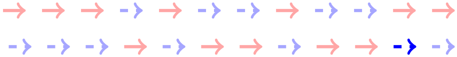
	1 → 2	2 → 3	3 → 1	4 → 4	1 ⇢ 2	2 ⇢ 4	3 ⇢ 3	4 ⇢ 1
A	**3**	3	3	3	3	2	3	2
B	3	3	3	**3**	2	3	3	2
C	3	**3**	3	3	3	3	1	3
D	3	3	**3**	3	2	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:

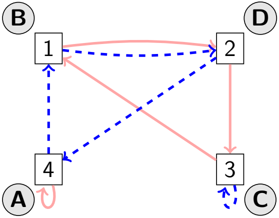


arcs visiting:

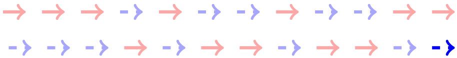
	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	3	3	3	3	3	**3**	3	2
B	3	3	3	3	2	3	3	**3**
C	3	3	3	3	3	3	**2**	3
D	3	3	3	3	**3**	2	3	3

Moving pebbles

Aim: Find a sequence of colors making each pebble visit each arc with same asymptotic frequency.



sequence of colors:



arcs visiting:

	1→2	2→3	3→1	4→4	1⇢2	2⇢4	3⇢3	4⇢1
A	3	3	3	3	3	3	3	**3**
B	3	3	3	3	**3**	3	3	3
C	3	3	3	3	3	3	**3**	3
D	3	3	3	3	3	**3**	3	3

Back to the results

Theorem G.–Meunier 2024+

If the graph is strongly connected, then there exists a sequence of colors making each pebble visit each arc with same asymptotic frequency.

2 proofs:

- Markov chains: non constructive
- Constructive proof with additionnal result:

Proposition

This balanced assignment is periodic of period bounded by $q^2 q!$

q: number of workers

Thank you!