Online Matchings for Graphs of Maximum Degree Three

Thomas Snow

University of Toronto

tsnow@cs.toronto.edu

Joint Work with Kanstantsin Pashkovich

Presentation Overview

Problem Example

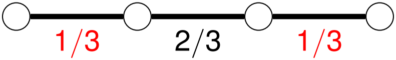
$$\sum_e y_e = 2/3$$

Maximum Matching $= 1$

Current Ratio $= 2/3$

Competitive Ratio $\leq 2/3$

Problem Example



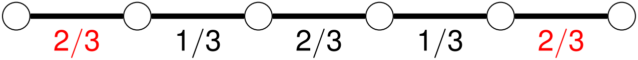
$$\sum_e y_e = 4/3$$

Maximum Matching $= 2$

Current Ratio $= 2/3$

Competitive Ratio $\leq 2/3$

Problem Example



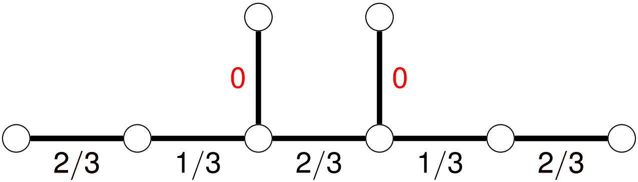
$$\sum_e y_e = 8/3$$

Maximum Matching $= 3$

Current Ratio $= 8/9$

Competitive Ratio $\leq \min\{2/3, 8/9\} = 2/3$

Problem Example



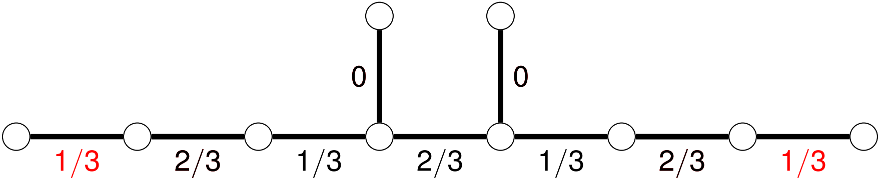
$$\sum_e y_e = 8/3$$

Maximum Matching $= 4$

Current Ratio $= 2/3$

Competitive Ratio $\leq 2/3$

Problem Example



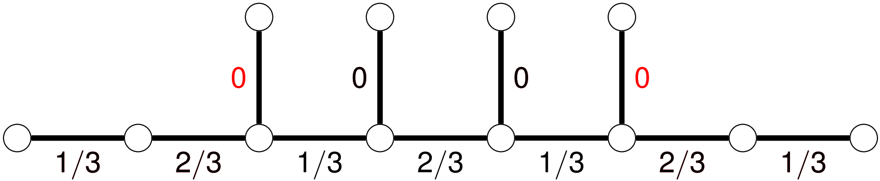
$$\sum_e y_e = 10/3$$

Maximum Matching $= 4$

Current Ratio $= 5/6$

Competitive Ratio $\leq \min\{2/3, 5/6\} = 2/3$

Problem Example



$$\sum_e y_e = 10/3$$

Maximum Matching $= 6$

Current Ratio $= 5/9 < 2/3$

Competitive Ratio $\leq \min\{2/3, 5/9\} = 5/9$

Goal

Given a target: $c \in [0, 1]$:

Goal

Design a strategy which assigns an arriving edge e a value $y_e \geq 0$ such that:

- Fractional matching constraints are satisfied: For all $u \in V$,
$$\sum_{f \in \delta(u)} y_f \leq 1$$
- At every time point:
$$\text{Current Ratio } := \frac{\sum_{f \in E} y_f}{|OPT_M|} \geq c$$

How large can c be?

Previous Work / Results

Known Bounds		
Maximum Degree	Lower Bound	Upper Bound
1	1.0	1.0
2	$2/3 \approx 0.667$ [BST'18]	$2/3 \approx 0.667$
3	$4/7 \approx 0.571$ [BST'18]	$4/(9-\sqrt{5}) \approx 0.5914$
4	$8/15 \approx 0.534$ [BST'18]	$4/(9-\sqrt{5}) \approx 0.5914$
d	$\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ [BST'18]	$\frac{1}{2}\left(1+\frac{1}{d+1}\right)$ [GKMSW'19]

Buchbinder, Segev, and Tkach 2018
Gamlath et al. 2019

Previous Work / Results

Known Bounds		
Maximum Degree	Lower Bound	Upper Bound
1	1.0	1.0
2	$2/3 \approx 0.667$ [BST'18]	$2/3 \approx 0.667$
3	$4/(9-\sqrt{5}) \approx 0.5914$	$4/(9-\sqrt{5}) \approx 0.5914$
4	$8/15 \approx 0.534$ [BST'18]	$4/(9-\sqrt{5}) \approx 0.5914$
d	$\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ [BST'18]	$\frac{1}{2}\left(1+\frac{1}{d+1}\right)$ [GKMSW'19]

Buchbinder, Segev, and Tkach 2018
Gamlath et al. 2019

Previous Work / Results

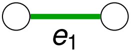
Known Bounds		
Maximum Degree	Lower Bound	Upper Bound
1	1.0	1.0
2	$2/3 \approx 0.667$ [BST'18]	$2/3 \approx 0.667$
3	$4/(9-\sqrt{5}) \approx 0.5914$	$4/(9-\sqrt{5}) \approx 0.5914$
4	$8/15 \approx 0.534$ [BST'18]	≈ 0.589
d	$\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ [BST'18]	$\frac{1}{2}\left(1+\frac{1}{d+1}\right)$ [GKMSW'19]

Buchbinder, Segev, and Tkach 2018
Gamlath et al. 2019

Consistent Instances

Arrival Order

Edge Types: 1 Path Edges



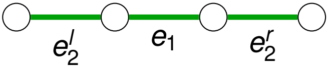
Theorem (Buchbinder, Segev, and Tkach 2018)

No Algorithm can achieve a competitive ratio larger than $c = 4/(9 - \sqrt{5}) \approx 0.5914$ *on graphs of maximum degree three.*

Consistent Instances

Arrival Order

Edge Types: 1. Path Edges



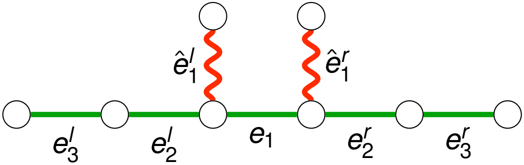
Theorem (Buchbinder, Segev, and Tkach 2018)

No Algorithm can achieve a competitive ratio larger than $c = 4/(9-\sqrt{5}) \approx 0.5914$ on graphs of maximum degree three.

Consistent Instances

Arrival Order

Edge Types: 1 Path Edges 2 Spokes



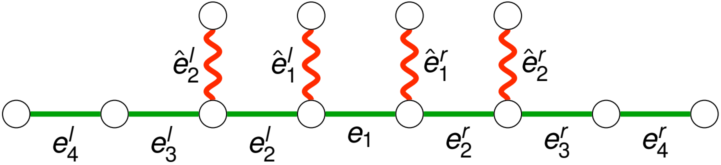
Theorem (Buchbinder, Segev, and Tkach 2018)

No Algorithm can achieve a competitive ratio larger than $c = 4/(9-\sqrt{5}) \approx 0.5914$ *on graphs of maximum degree three.*

Consistent Instances

Arrival Order

Edge Types: 1 Path Edges 2 Spokes



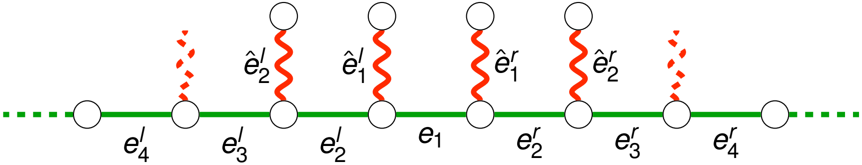
Theorem (Buchbinder, Segev, and Tkach 2018)

No Algorithm can achieve a competitive ratio larger than $c = 4/(9 - \sqrt{5}) \approx 0.5914$ *on graphs of maximum degree three.*

Consistent Instances

Arrival Order

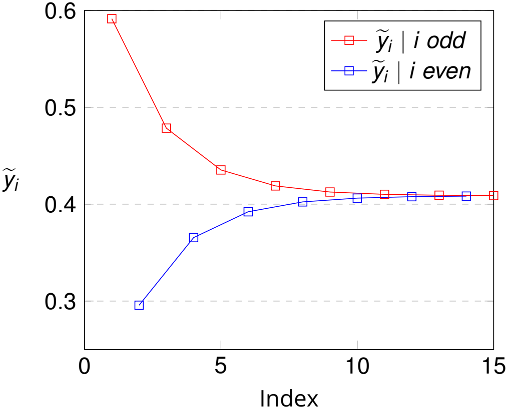
Edge Types: 1 Path Edges 2 Spokes



Theorem (Buchbinder, Segev, and Tkach 2018)

No Algorithm can achieve a competitive ratio larger than $c = 4/(9-\sqrt{5}) \approx 0.5914$ *on graphs of maximum degree three.*

Ideal Assignments



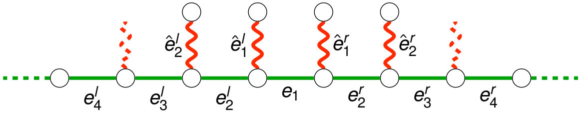
Recall: $c \approx 0.5914$

$\widetilde{y}_1 := c$

$\widetilde{y}_2 := \frac{c}{2}$

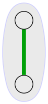
$\widetilde{y}_3 := \frac{5c-2}{2}$

$\widetilde{y}_n := c-(1-\widetilde{y}_{n-1}-\widetilde{y}_{n-2})$

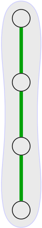


How do we handle general arrival orders?

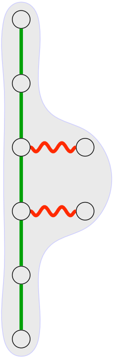
Online Partitioning



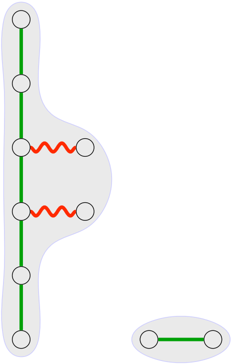
Online Partitioning



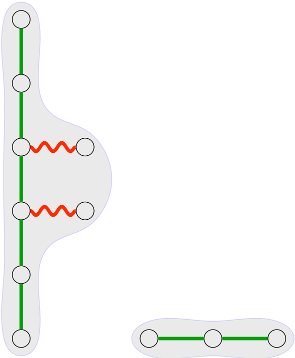
Online Partitioning



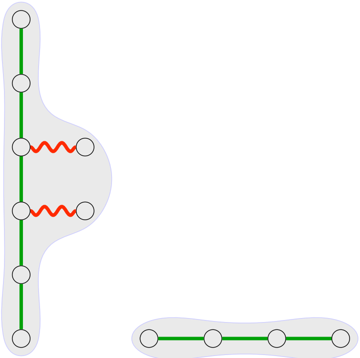
Online Partitioning



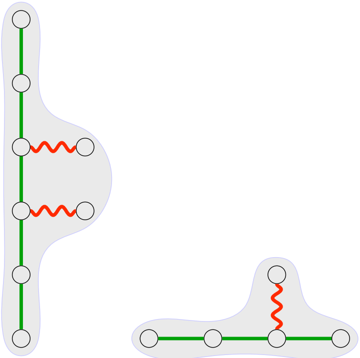
Online Partitioning



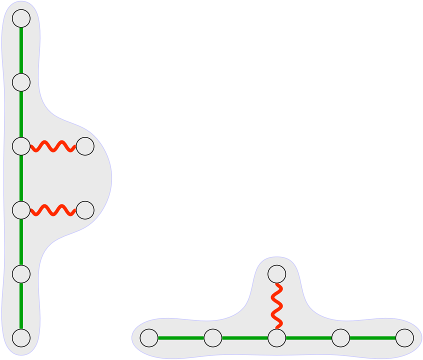
Online Partitioning



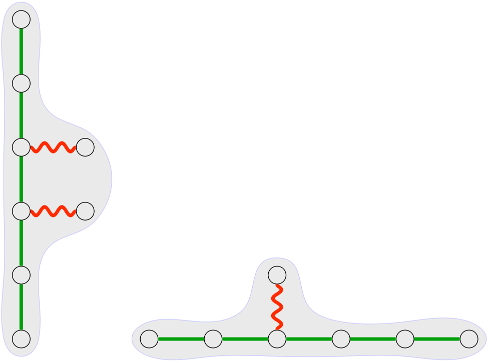
Online Partitioning



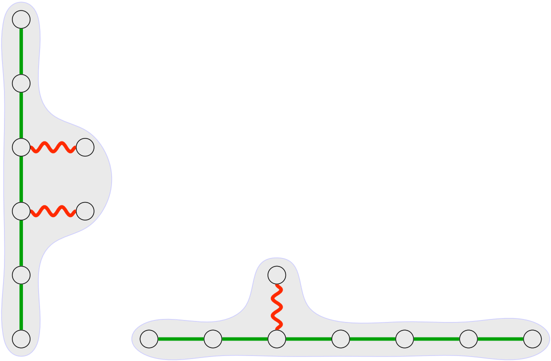
Online Partitioning



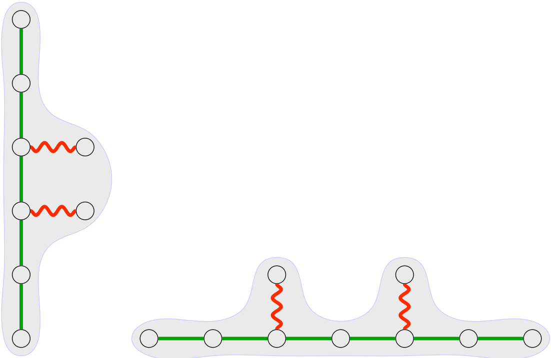
Online Partitioning



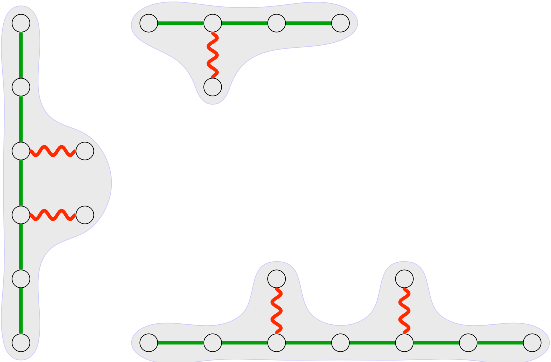
Online Partitioning



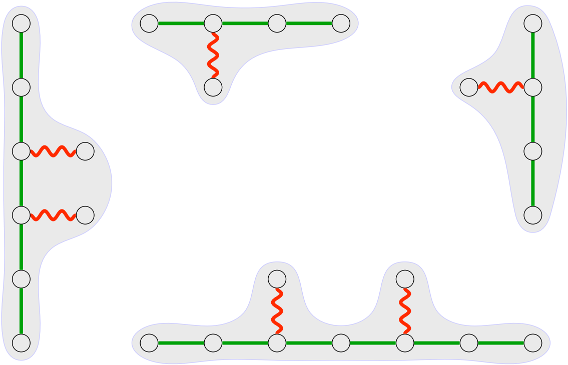
Online Partitioning



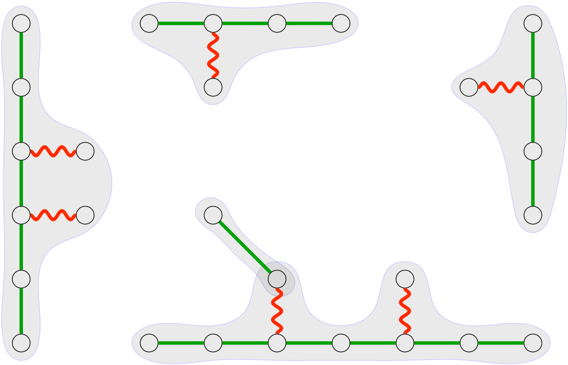
Online Partitioning



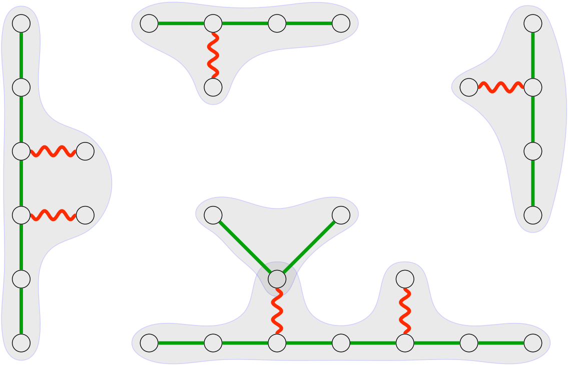
Online Partitioning



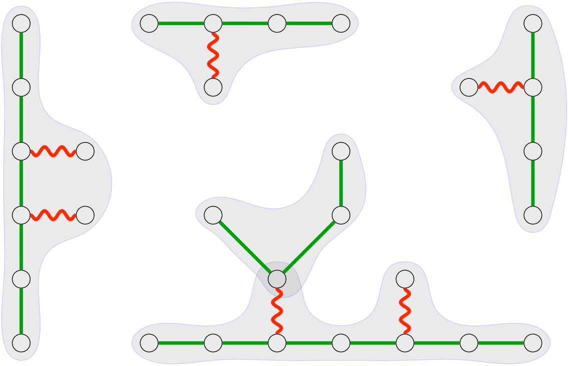
Online Partitioning



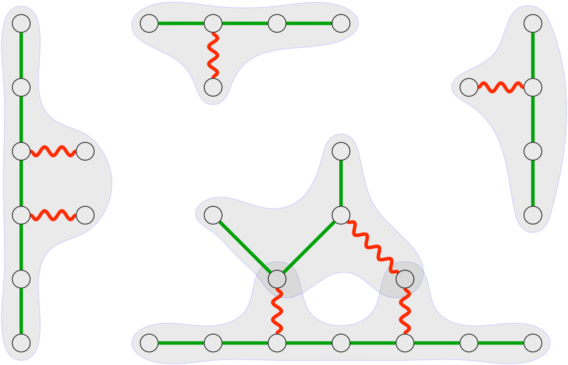
Online Partitioning



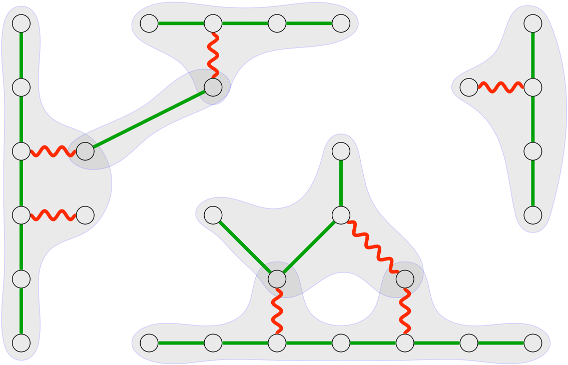
Online Partitioning



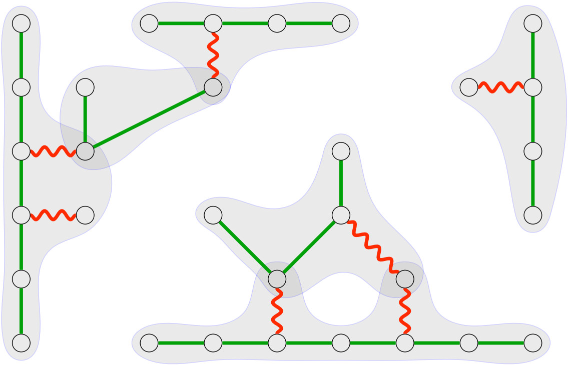
Online Partitioning



Online Partitioning

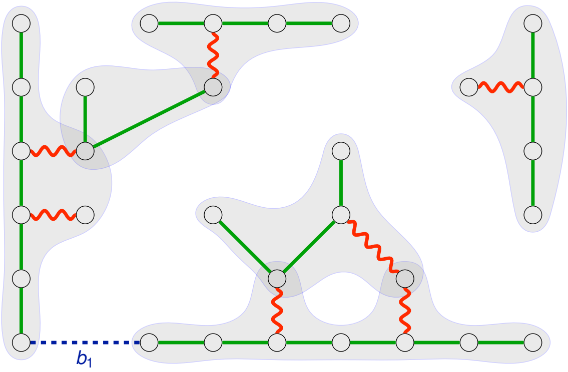


Online Partitioning



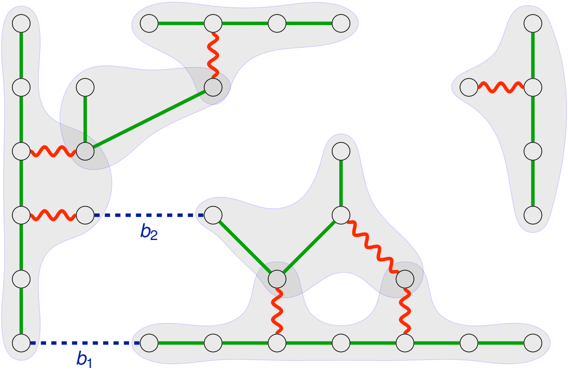
Online Partitioning

New Edge Type: 3 Bridges



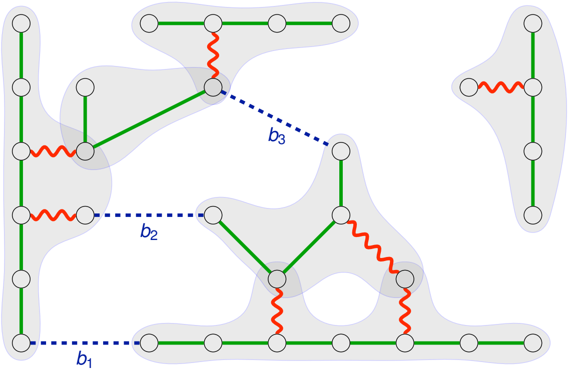
Online Partitioning

New Edge Type: 3 Bridges



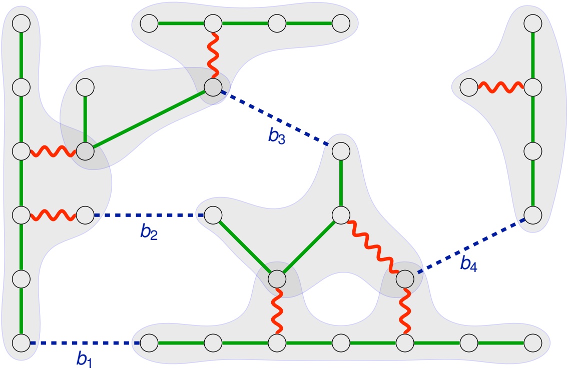
Online Partitioning

New Edge Type: ③ Bridges



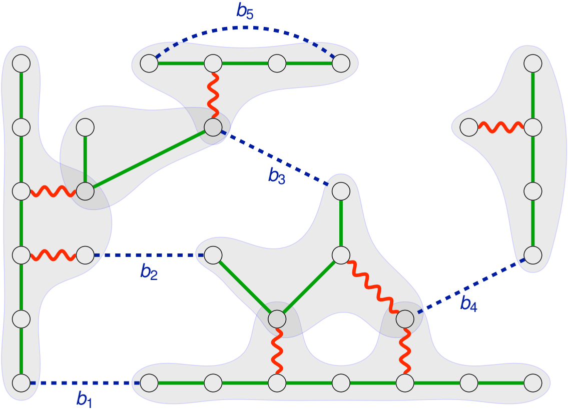
Online Partitioning

New Edge Type: 3 Bridges



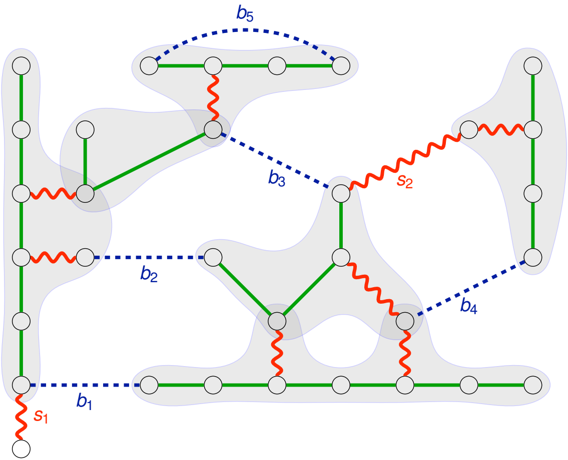
Online Partitioning

New Edge Type: 3 Bridges



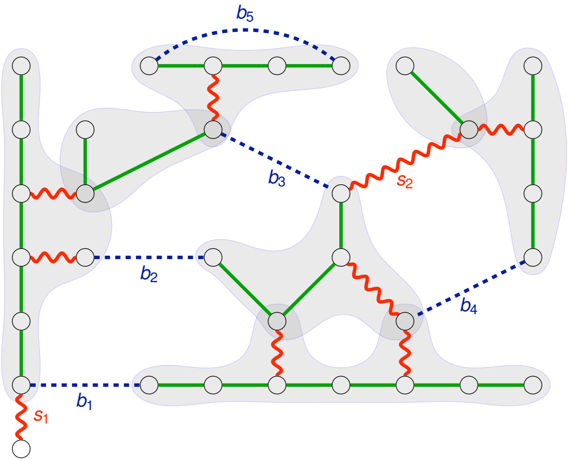
Online Partitioning

New Edge Type: 3 Bridges



Online Partitioning

New Edge Type: 3 Bridges



Classifying an Arriving Edge

Arriving edge $e = uv$

Question

Should we assign e a path edge, a spoke, or a bridge?

Question

Should e join an existing consistent instance? If so, which one?

Question

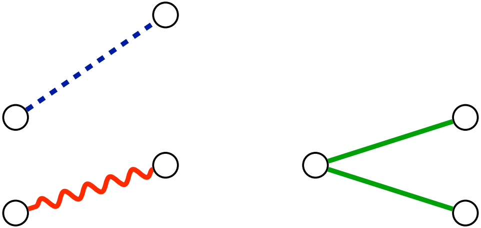
What value should we assign e?

Classifying an Arriving Edge (Continued)

Definition

For $A \subseteq E$, $\text{type}(A) := (t_1, t_2, t_3)$ where,

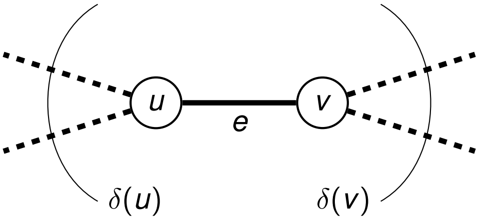
- $t_1 = \#$ path edges in A
- $t_2 = \#$ spokes in A
- $t_3 = \#$ bridges in A



$\text{type}(\cdot) = (2, 1, 1)$

Classifying an Arriving Edge (Continued)

Arriving edge $e = uv$



Consider type $(\delta(u))$ and type $(\delta(v))$

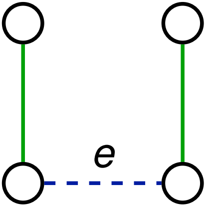
Classifying an Arriving Edge

Bridges

When should we assign e a bridge?

Classifying an Arriving Edge

Bridges

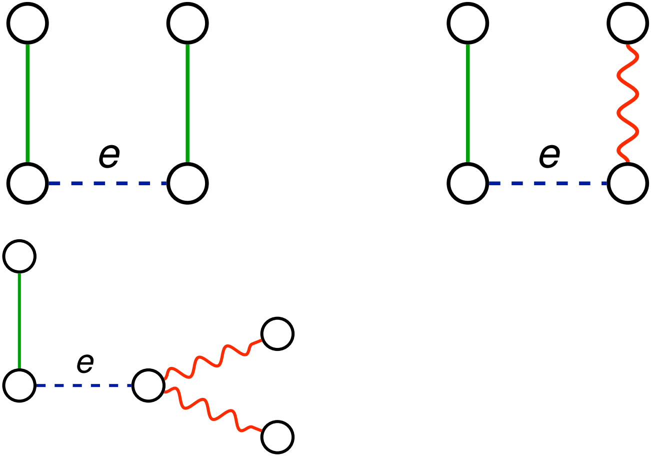


Classifying an Arriving Edge

Bridges

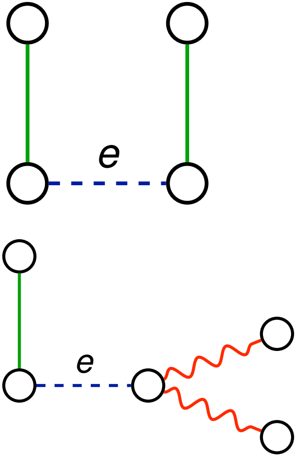
Classifying an Arriving Edge

Bridges



Classifying an Arriving Edge

Bridges



Classifying an Arriving Edge

If e is not a bridge, what should we do?

Classifying an Arriving Edge

Paths and Spokes

Arriving edge $e = uv$
Assume $\deg(u) \geq \deg(v)$

If $e = uv$ is not a bridge...

1. Choose an endpoint $z(e) \in \{u, v\}$

$$z(e) \leftarrow \begin{cases} v & \text{if } \deg(v) = 3 \;\&\; \sum_{f \in \delta(v)\setminus\{e\}} y_f > \sum_{f \in \delta(u)\setminus\{e\}} y_f \\ u & \text{otherwise} \end{cases}$$

2. Look at $\deg(z(e))$ and $\text{type}\left(\delta_{G\setminus e}(z(e))\right)$

If $\deg(z(e)) = 3$, when should e not be assigned a spoke?

- If $\text{type}\left(\delta_{G\setminus e}(z(e))\right) = (0, 2, 0)$ or $(1, 1, 0)$

Classifying an Arriving Edge

Paths and Spokes

If $\deg(z(e)) < 3$ or type $\left(\delta_{G\setminus e}(z(e))\right) = (0, 2, 0)$ or $(1, 1, 0)$

- e gets assigned a path

Either

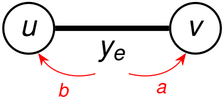
- e joins an existing consistent instance
  - e extends an existing path. i.e. there is a path edge in $\delta_{G\setminus e}(z(e))$
- e starts a new consistent instance

Feasibility and Competitiveness

Lemma (Correctness Properties)

1. $\forall\ u \in V\ \sum_{e\in\delta(u)} y_e \leq 1$
2. $\forall\ e = uv \in E\ x_u + x_v \geq c$
3. $\sum_{e\in E} y_e = \sum_{u\in V} x_u$

To Accomplish Property 3: For Arriving Edge $e = uv$



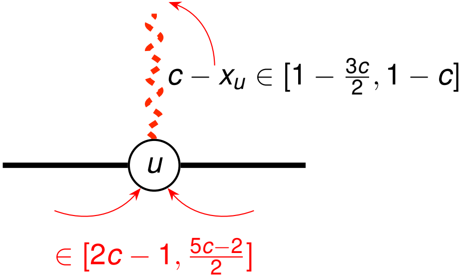
Require: $a + b = y_e$
Note: a or b can be negative.

Cover Structure

Lemma (Cover Properties (Spokes))

1. $\forall\, u \in V$ *with* $deg(u) = 2$ *type* $(\delta(u)) \notin \{(0,2,0),(1,1,0)\}$

$$x_u \in \left[2c-1, \frac{5c-2}{2}\right]$$

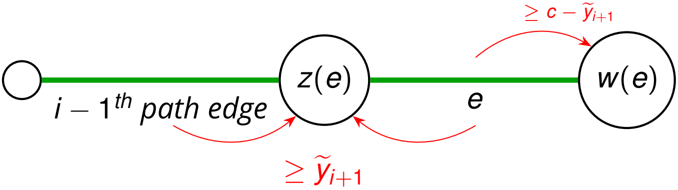


Cover Structure

Lemma (Cover Properties (Paths))

1. $\forall$ *path edges* $e = uv$, *if* e *is the* i^{th} *edge in the path then,*

$$x_{z(e)} \geq \widetilde{y}_{i+1} \quad \text{and} \quad x_{w(e)} \geq c - \widetilde{y}_{i+1}$$



Open Problem

- Randomized Integral Algorithms for graphs of maximum degree $d > 2$?
- Does there exist a randomized Integral algorithm for bipartite graphs of maximum degree 3 achieving c-competitiveness?
- What's the minimum d for which on graphs of maximum degree d, vertex arrivals are strictly easier than edge arrivals?
    - Wang and Wong 2015 & Gamlath et al. 2019 show that for general graphs, vertex arrivals are easier than edge arrivals.
    - $d \neq 2$, Buchbinder, Segev, and Tkach 2018
    - $d \neq 3$, Buchbinder, Segev, and Tkach 2018 & this work

References

Buchbinder, Niv, Danny Segev, and Yevgeny Tkach (Aug. 2018). "Online algorithms for maximum cardinality matching with edge arrivals". In: *Algorithmica* 81.5, pp. 1781–1799. DOI: 10.1007/s00453-018-0505-7.

Gamlath, Buddhima et al. (2019). "Online Matching with General Arrivals". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 26–37. DOI: 10.1109/FOCS.2019.00011.

Wang, Yajun and Sam Chiu-wai Wong (2015). "Two-sided Online Bipartite Matching and Vertex Cover: Beating the Greedy Algorithm". In: *Automata, Languages, and Programming*. Ed. by Magnús M. Halldórsson et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1070–1081. ISBN: 978-3-662-47672-7. DOI: 10.1007/978-3-662-47672-7_87.