New Algorithms, Better Bounds, and a Novel Model for Online Stochastic Matching

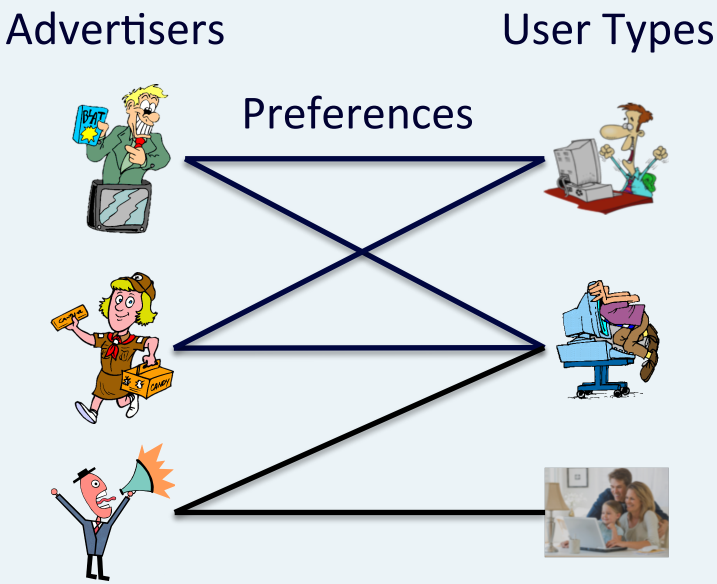
Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu

University of Maryland, College Park

European Symposium on Algorithms – ESA 2016

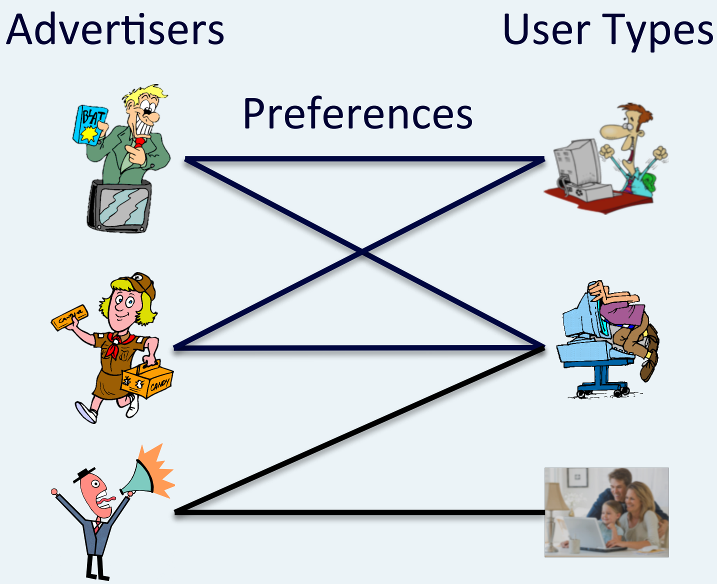
Online Matching: Known I.I.D. Model

- Input: bipartite graph $G = (U, V, E)$
  - U is set of offline vertices
  - V is a set of online vertex *types*



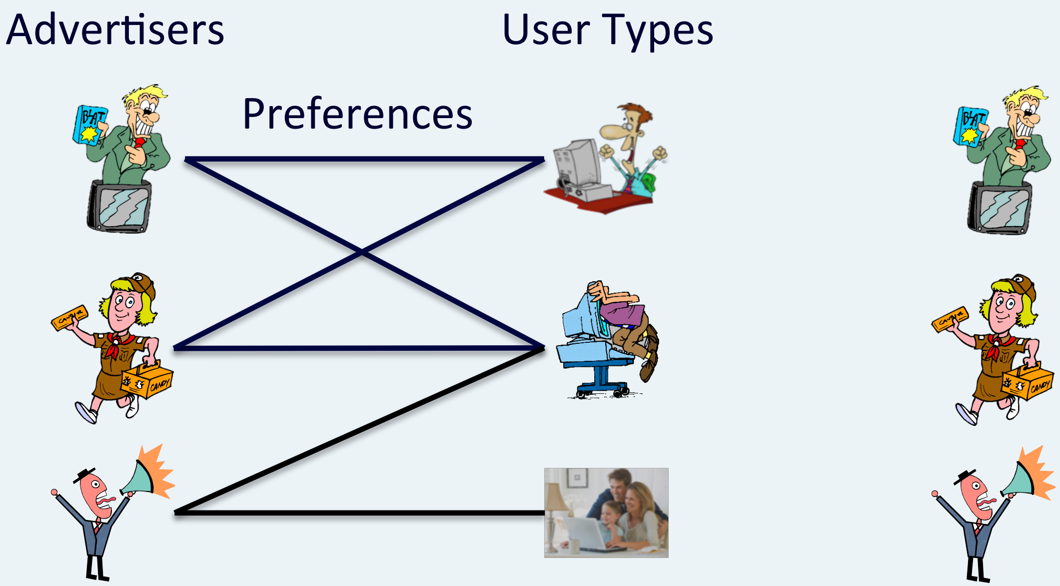
Online Matching: Known I.I.D. Model

- Arrivals i.i.d. from a known distribution on *V*
  - Each arrival of some *v* in *V* is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)



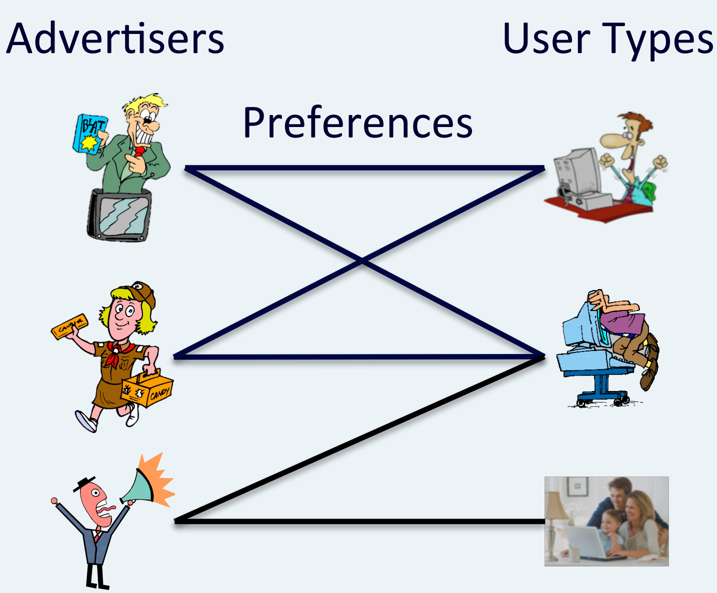
Online Matching: Known I.I.D. Model

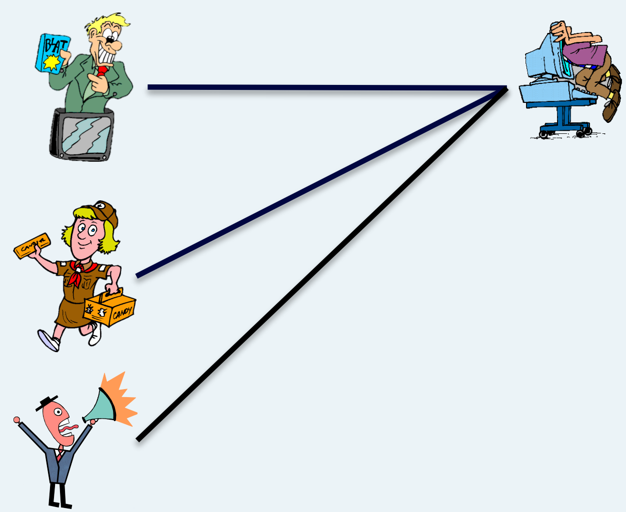
- Arrivals i.i.d. from a known distribution on *V*
  - Each arrival of some *v* in *V* is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)



Online Matching: Known I.I.D. Model

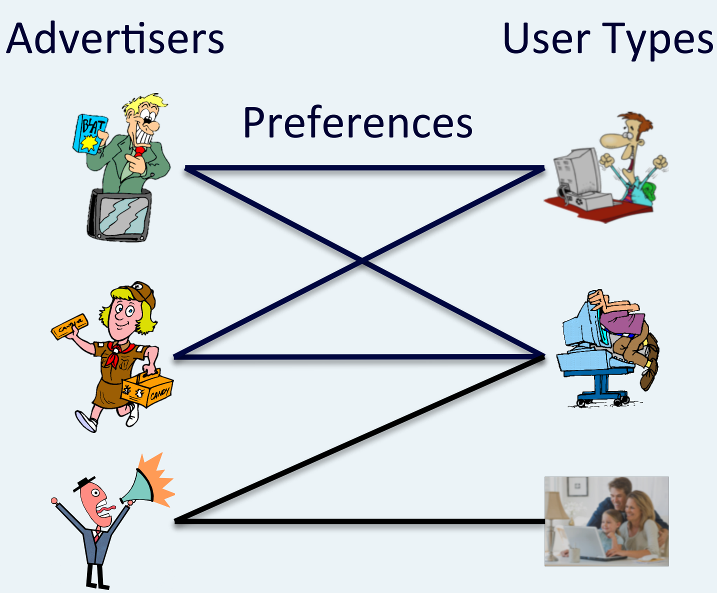
- Arrivals i.i.d. from a known distribution on *V*
  - Each arrival of some *v* in *V* is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

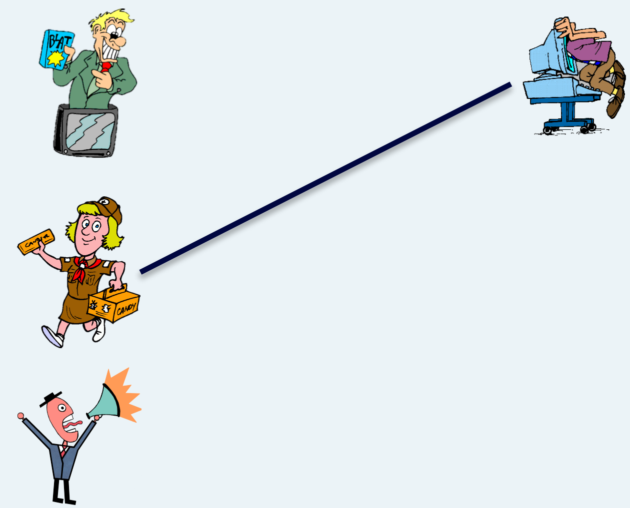




Online Matching: Known I.I.D. Model

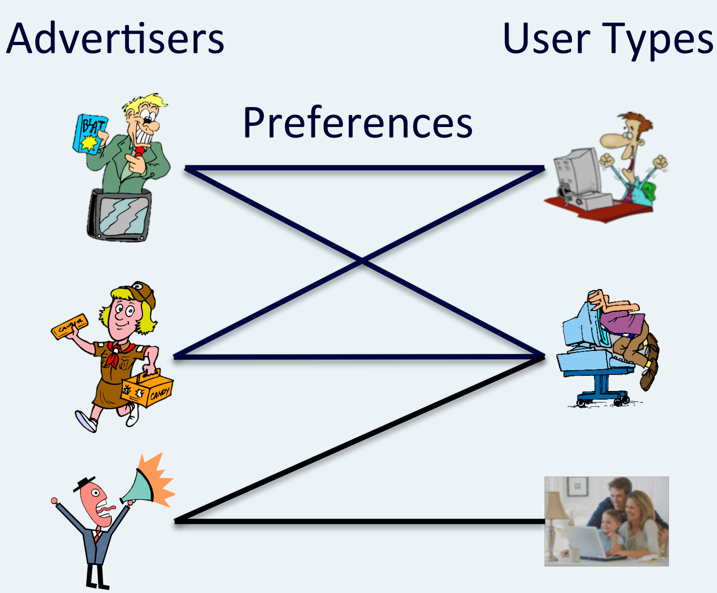
- Arrivals i.i.d. from a known distribution on V
  - Each arrival of some v in V is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

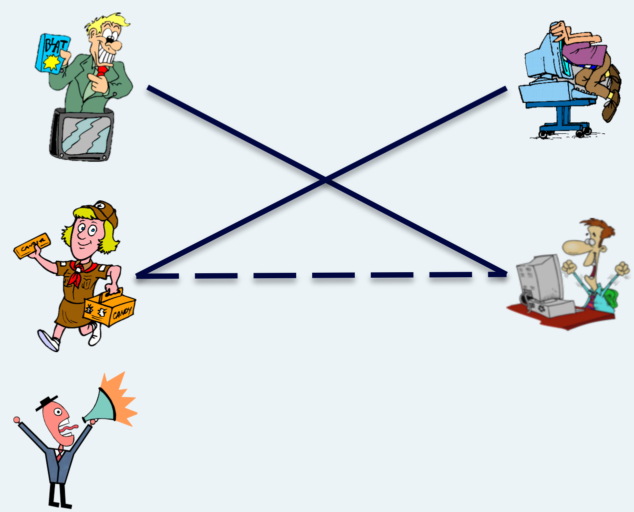




Online Matching: Known I.I.D. Model

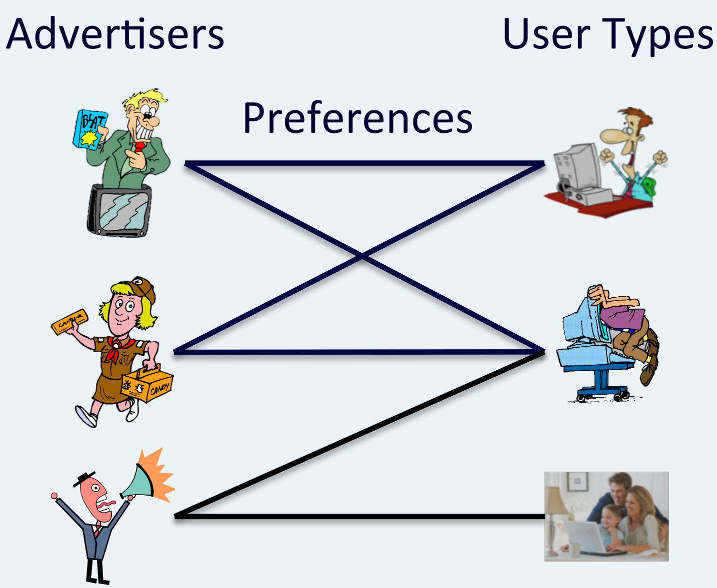
- Arrivals i.i.d. from a known distribution on V
  - Each arrival of some v in V is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

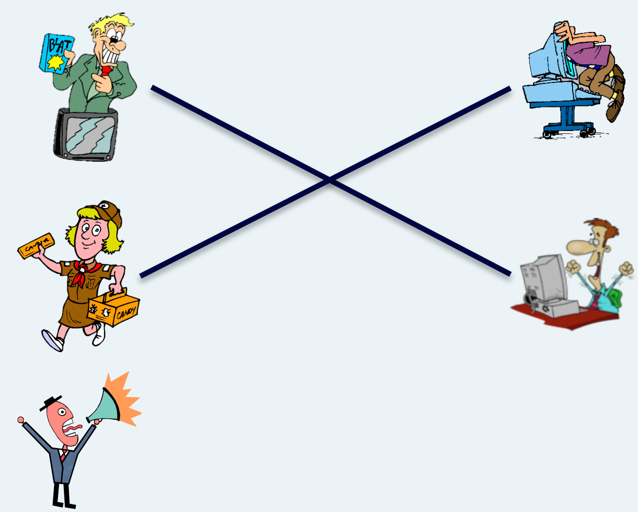




Online Matching: Known I.I.D. Model

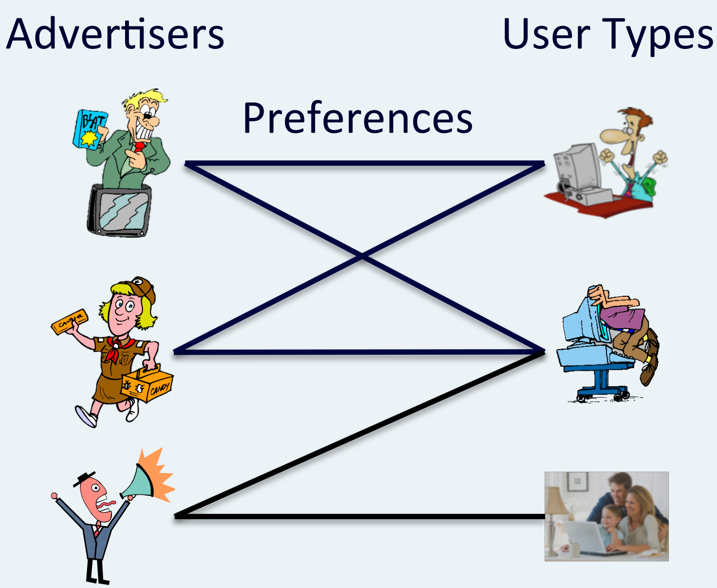
- Arrivals i.i.d. from a known distribution on V
  - Each arrival of some v in V is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

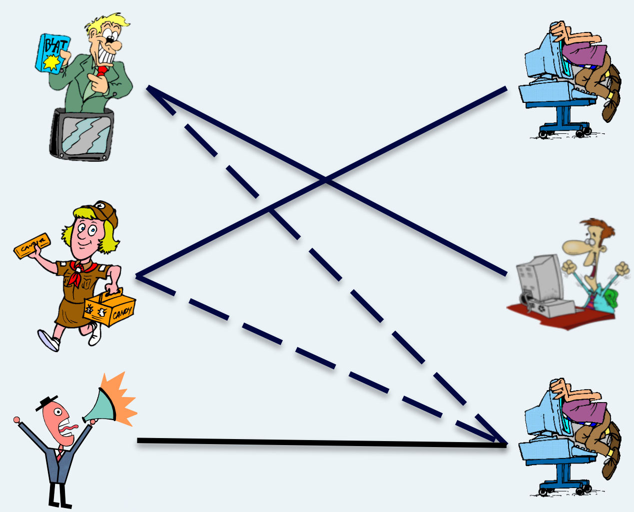




Online Matching: Known I.I.D. Model

- Arrivals i.i.d. from a known distribution on V
  - Each arrival of some v in V is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

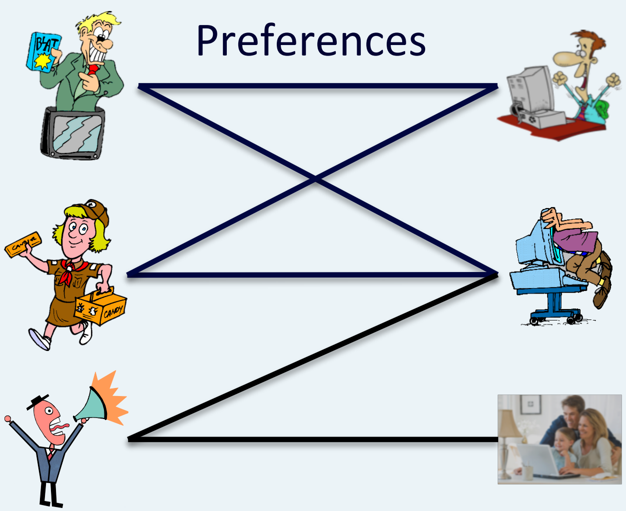


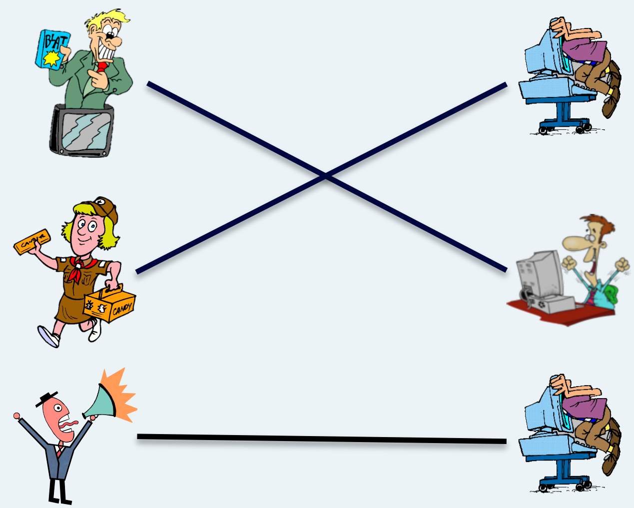


Online Matching: Known I.I.D. Model

- Arrivals i.i.d. from a known distribution on V
  - Each arrival of some v in V is a distinct vertex
  - WLOG uniform distribution (integral arrival rates assumed)

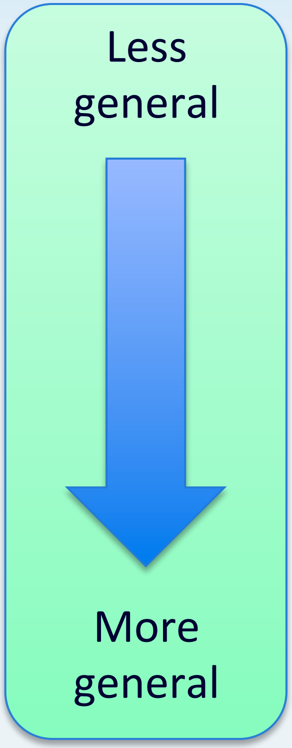
Advertisers User Types





Variants

- Unweighted
- Vertex-weighted
  - weights only on offline vertices *U*
- Edge-weighted
- Stochastic rewards
  - Edges exist (independently) with given probabilities



Competitive Ratio

- ALG = any online algorithm
- OPT = optimal online algorithm
- Competitive ratio = **E**[ALG]/**E**[OPT]

Related and Our Work

Variant	Ratio	Authors
Unweighted	0.67	Feldman, Mehta, Mirokkni, Muthukrishnan (FOCS 2009)
	0.705	Manshadi, Oveis, Gharan, Saberi (SODA 2011)
	0.7293	Jaillet and Lu (Math OR 2013)
	0.7299	This work (ESA 2016)
Vertex-weighted	0.725	Jaillet and Lu (Math OR 2013)
	0.7299	This work (ESA 2016)
Edge-weighted	0.667	Haeupler, Mirokkni, Zadimoghaddam (WINE 2011)
	0.705	This work (ESA 2016)
Stochastic Rewards	0.632	This work (ESA 2016)

Two Phases: Offline & Online

- Offline phase:
  - Preprocess input graph and develop guidelines for the online phase
  - Primary focus of contributions presented here
- Online Phase:
  - Vertices arrive and must be matched to an offline neighbor or discarded

Benchmark LP

$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} w_e f_e \\
\text{subject to} \quad & \sum_{e \in \partial(u)} f_e \leq 1 && \forall u \in U \\
& \sum_{e \in \partial(v)} f_e \leq 1 && \forall v \in V \\
& 0 \leq f_e \leq 1 - 1/e && \forall e \in E
\end{aligned}
$$

Matching constraints

Assuming arrival rate is 1

Probability some *v* never arrives = 1/e
Expected value of any edge ≤ 1-1/e

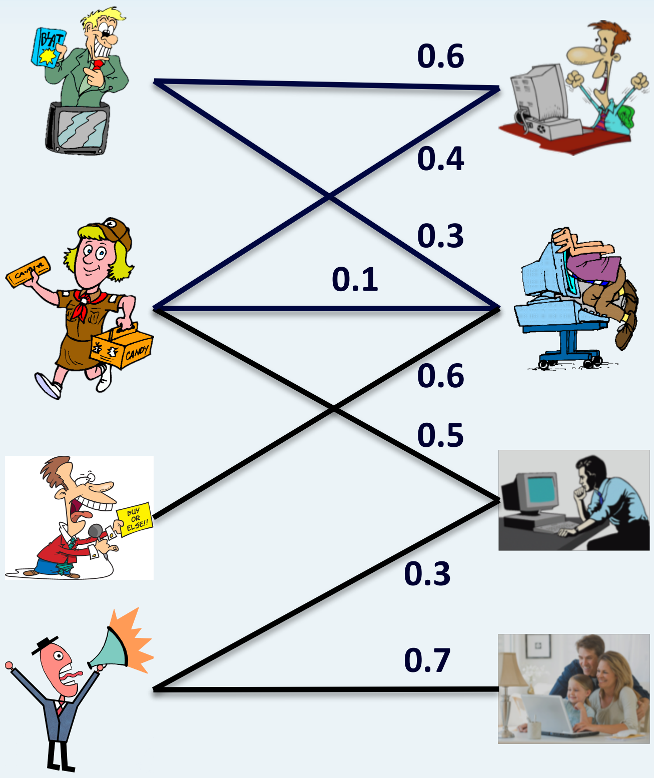
Scaled Dependent Rounding

- Generalizes dependent rounding due to Gandhi, Khuller, Parthasarathy, and Srinivasan (FOCS 2004)
  - Oft called GKPS rounding
- Multiply LP solution by k before rounding
- Properties of dependent rounding still hold

1. **Marginal distribution:** For every edge e, let $p_e = kf_e - \lfloor kf_e \rfloor$. Then, $\Pr[F_e = \lceil kf_e \rceil] = p_e$ and $\Pr[F_e = \lfloor kf_e \rfloor] = 1 - p_e$.
2. **Degree-preservation:** For any vertex $w \in U \cup V$, let its fractional degree kf_w be $\sum_{e \in \partial(w)} kf_e$ and integral degree be the random variable $F_w = \sum_{e \in \partial(w)} F_e$. Then $F_w \in \{\lfloor kf_w \rfloor, \lceil kf_w \rceil\}$.

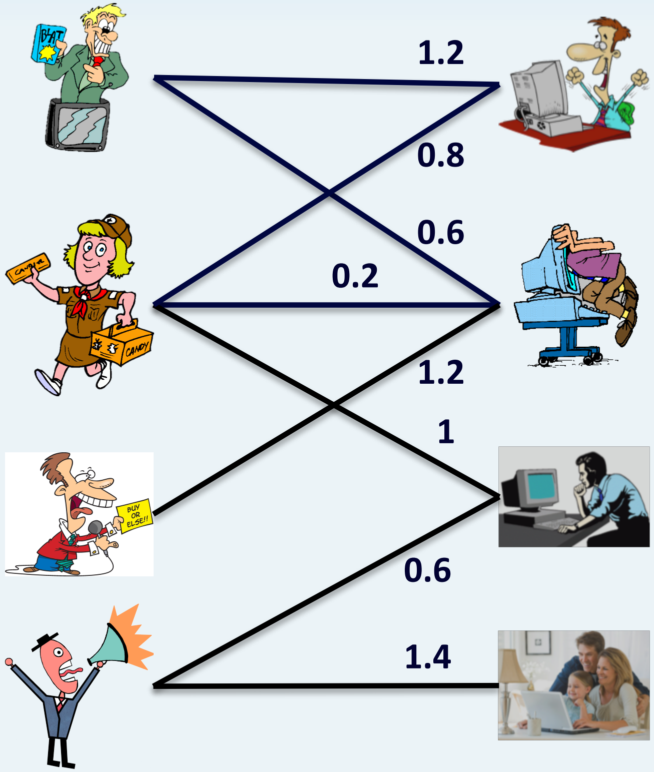
Edge-weighted

Warm-up Algorithm: Offline phase



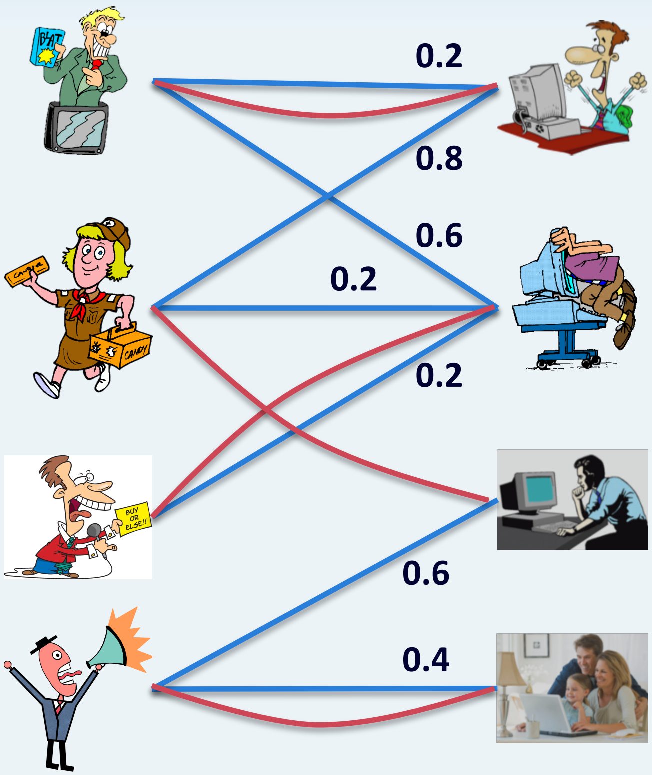
- Assign LP values to edges

Warm-up Algorithm: Offline phase



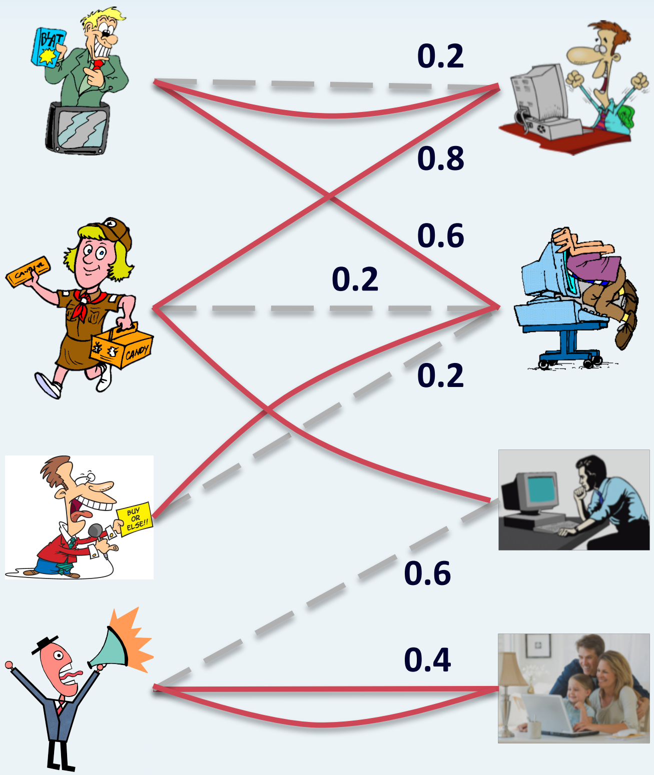
- Assign LP values to edges
- Multiply values by 2

Warm-up Algorithm: Offline phase



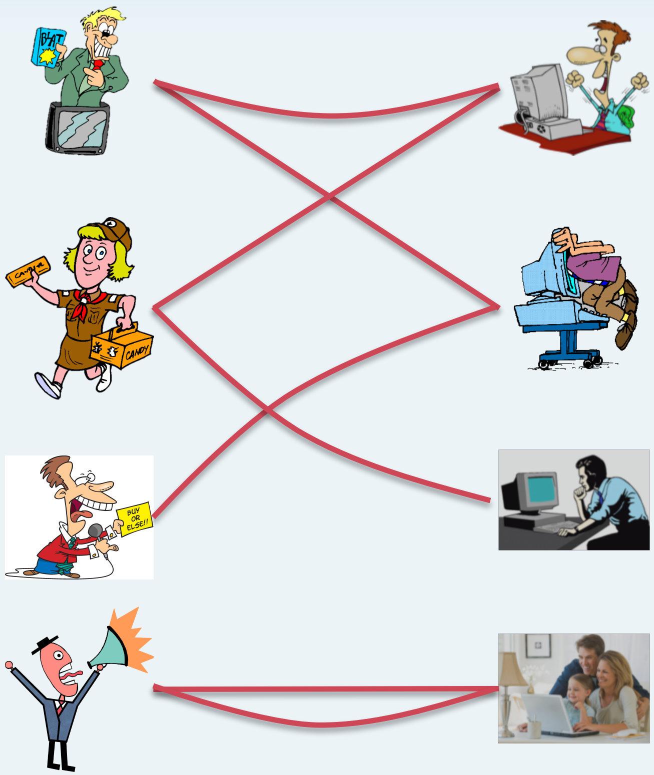
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts

Warm-up Algorithm: Offline phase



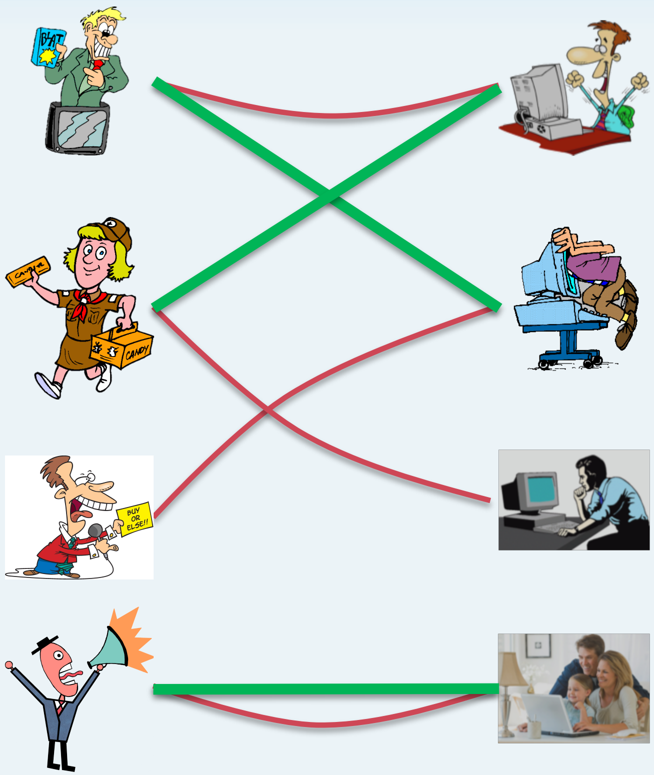
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts
- Apply dependent rounding

Warm-up Algorithm: Offline phase



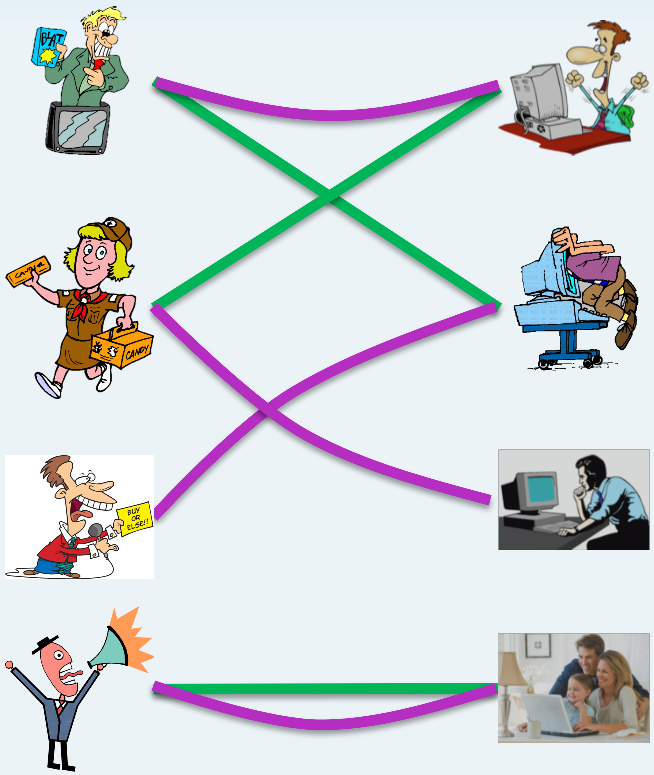
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts
- Apply dependent rounding

Warm-up Algorithm: Offline phase



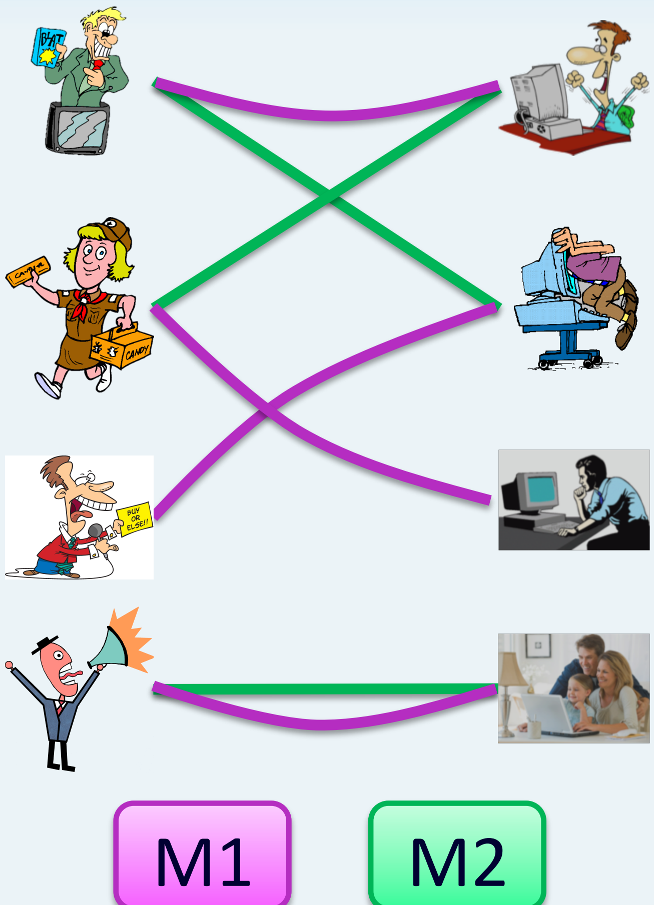
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts
- Apply dependent rounding
- Decompose into 2 matchings

Warm-up Algorithm: Offline phase



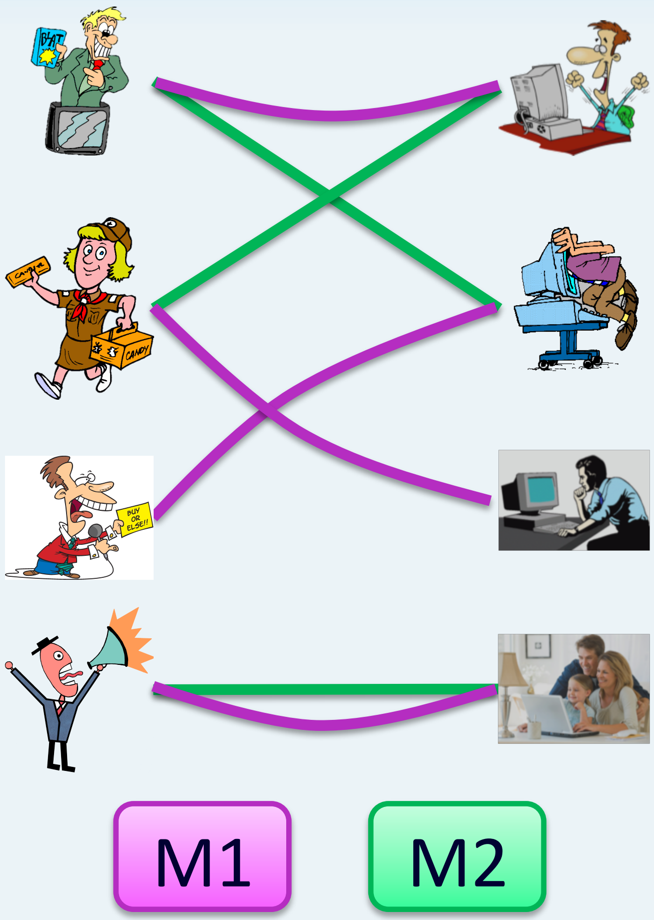
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts
- Apply dependent rounding
- Decompose into 2 matchings

Warm-up Algorithm: Offline phase



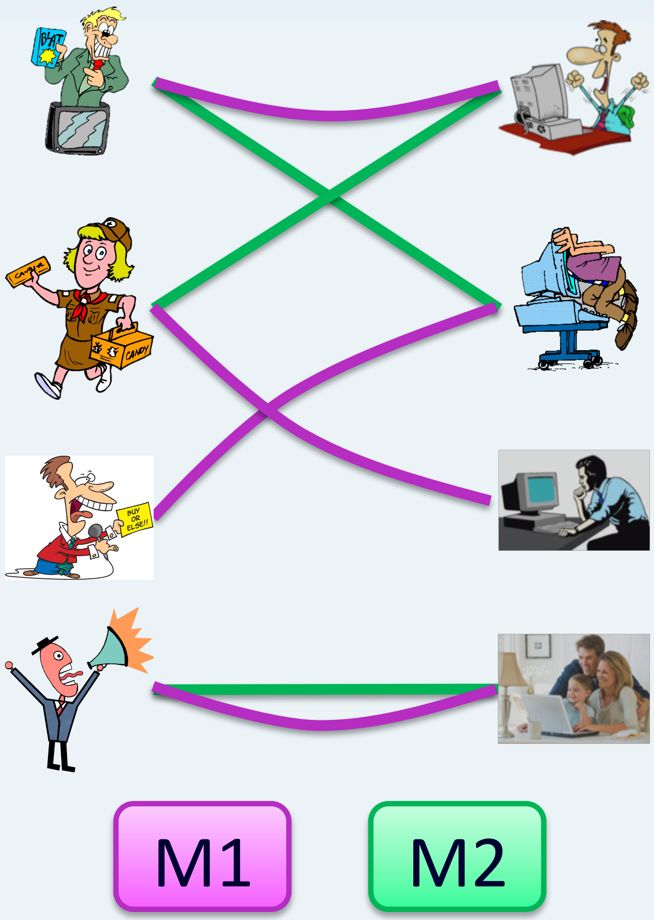
- Assign LP values to edges
- Multiply values by 2
- Separate integral and fractional parts
- Apply dependent rounding
- Decompose into 2 matchings
- Randomly order the matchings M1 and M2

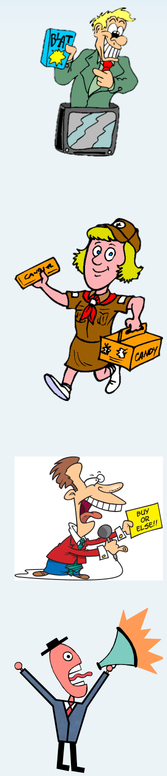
Warm-up Algorithm: Online phase



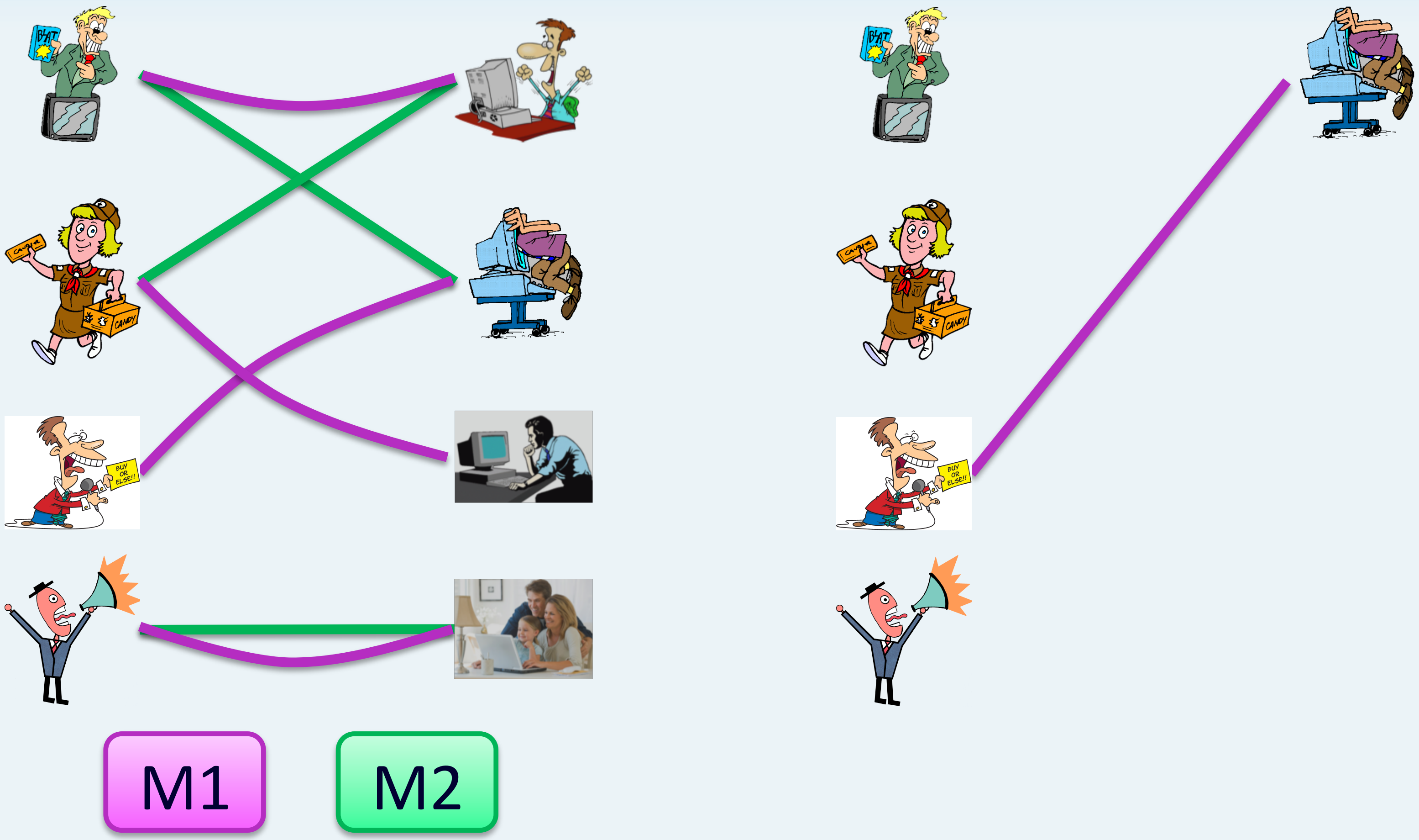
- First arrival of a type
  - Attempt match to M1 neighbor
- Second arrival
  - Attempt match to M2 neighbor
- Third or later arrival
  - Do nothing

Warm-up Algorithm: Online phase

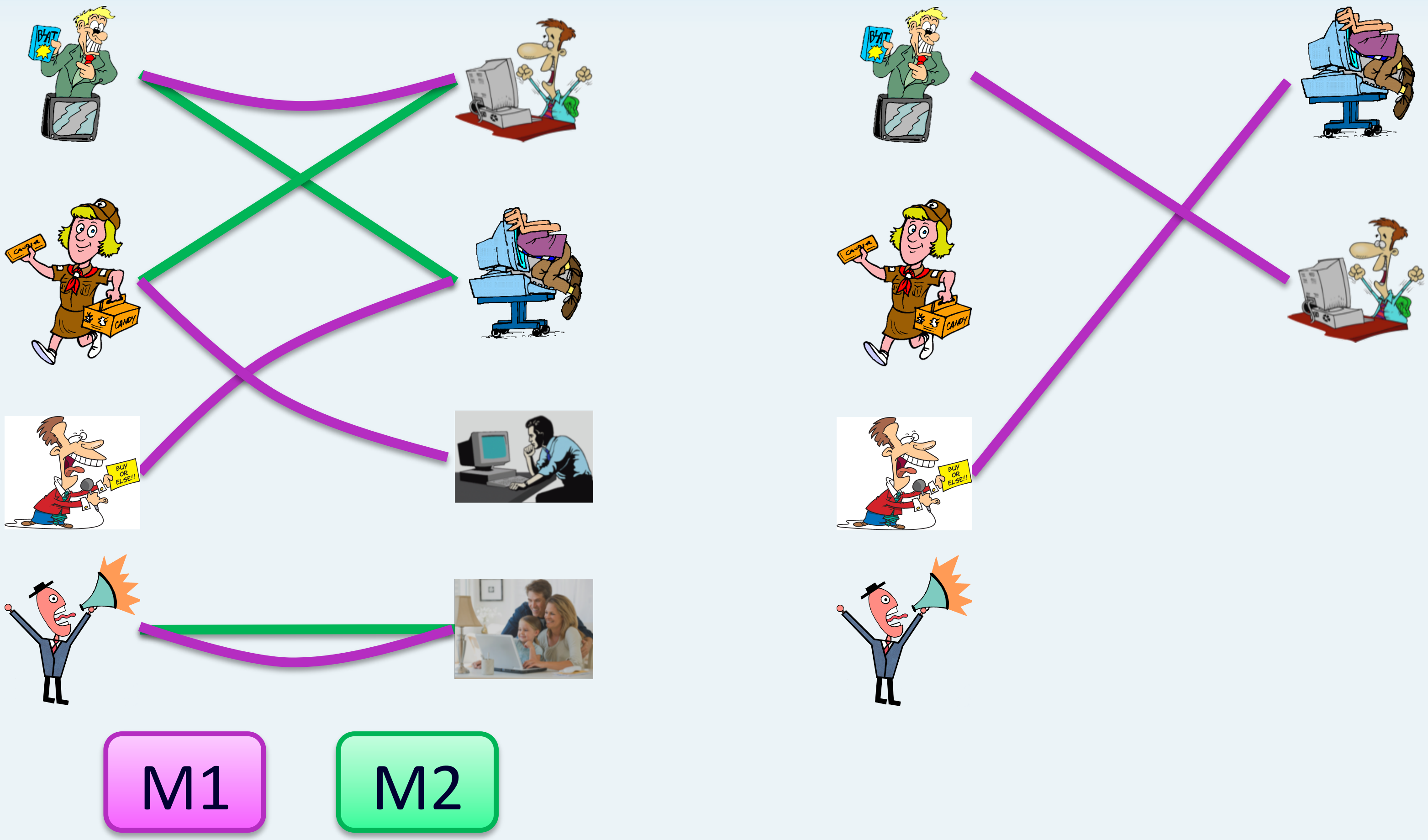




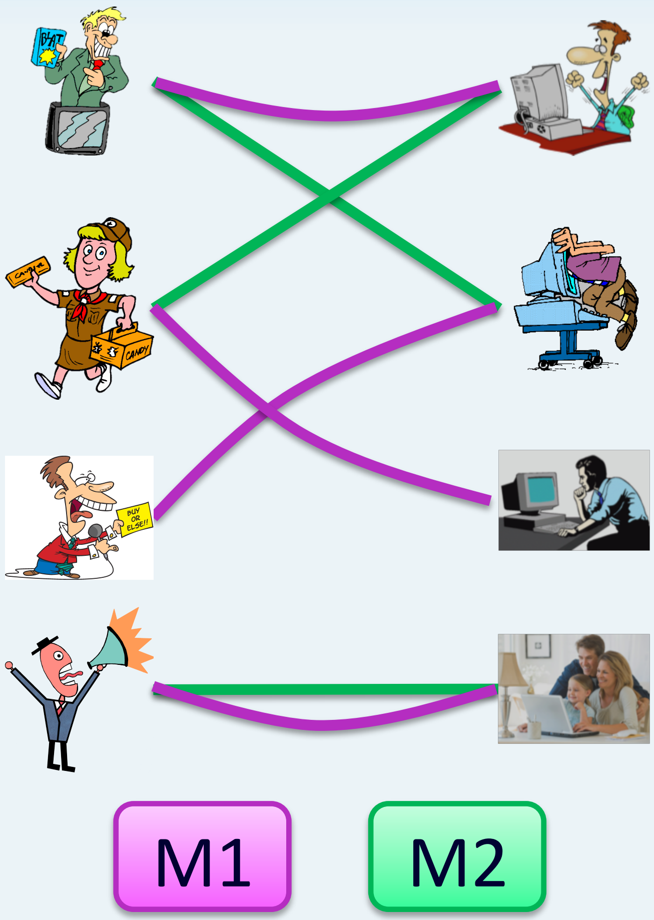
Warm-up Algorithm: Online phase

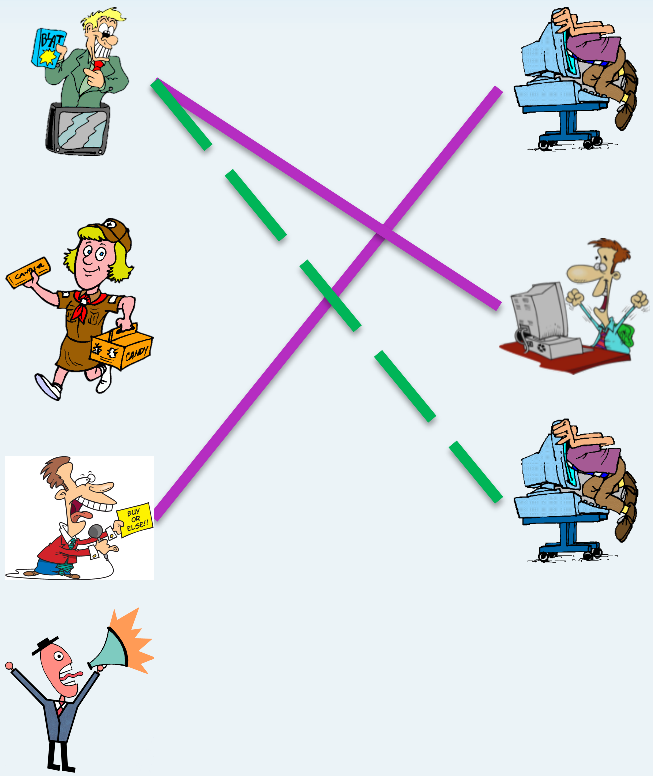


Warm-up Algorithm: Online phase

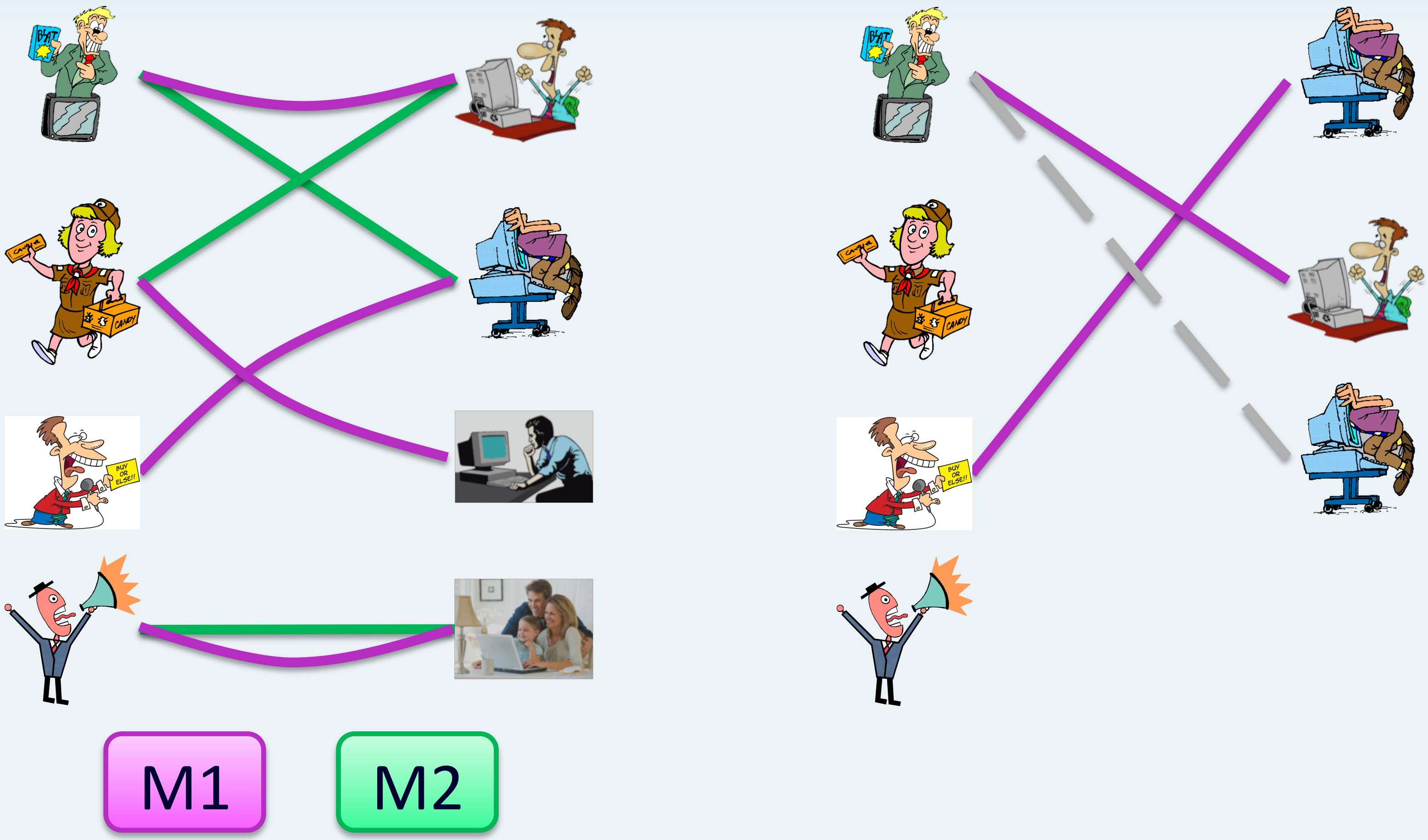


Warm-up Algorithm: Online phase

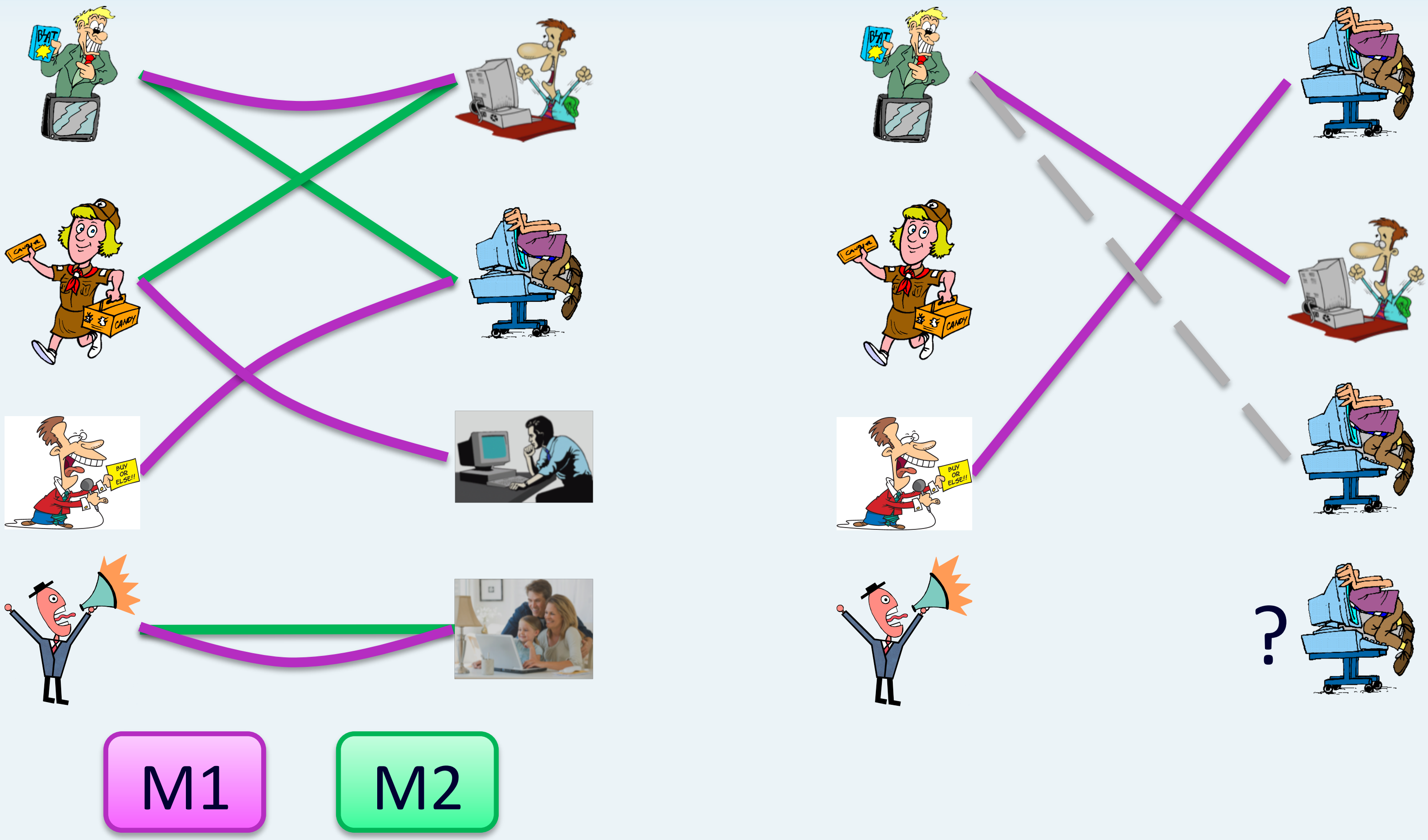




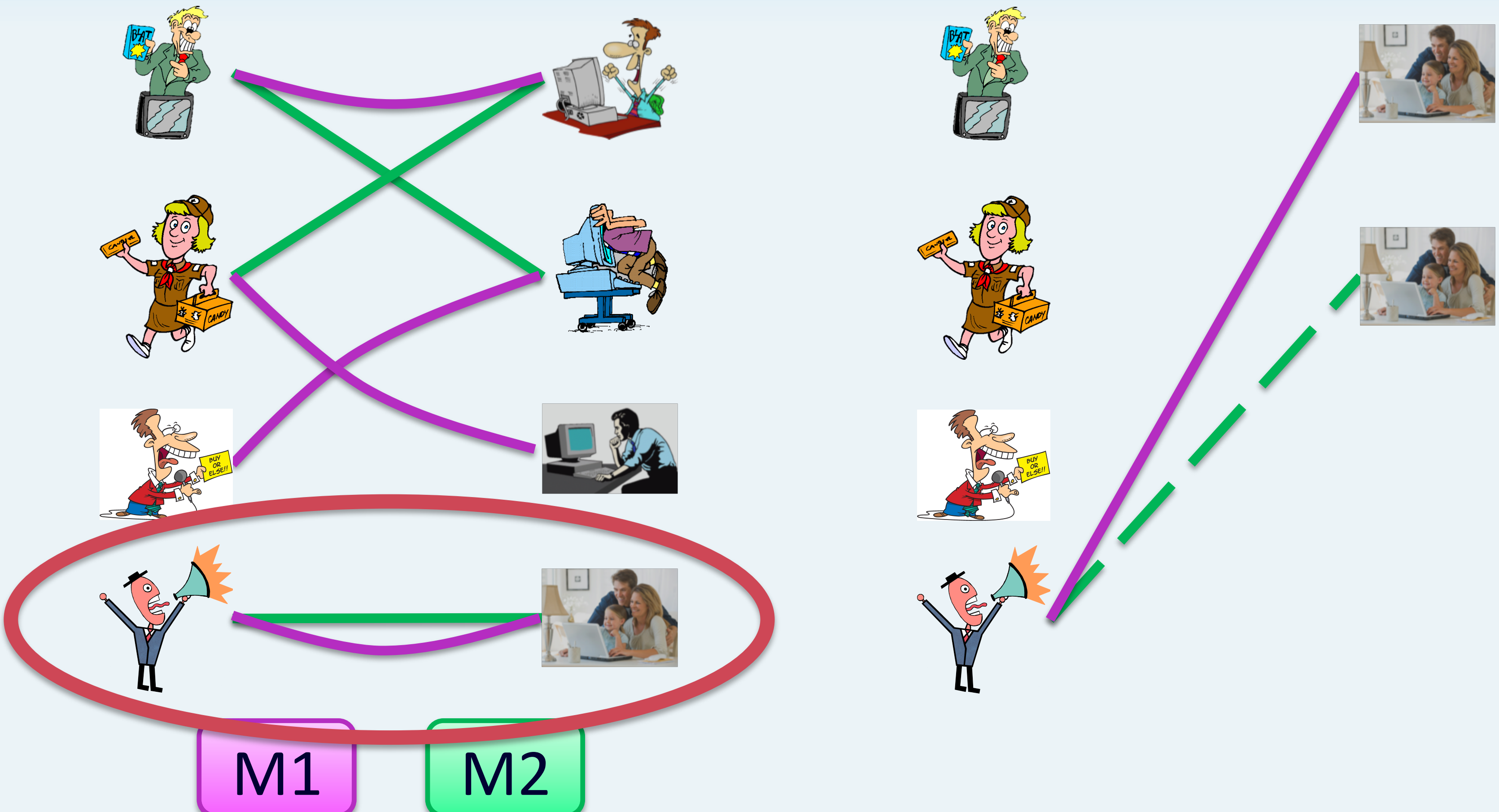
Warm-up Algorithm: Online phase



Warm-up Algorithm: Online phase



Challenge: Two Copies of an Edge



Intuition behind Improvement

- If an edge appears in both matchings, the second arrival is wasted
- Haeupler *et al* (2011) also used two matchings
  - Matchings chosen independently
  - Probability edges appear in both matchings ≤ 0.63

We show correlated rounding reduces this probability to ≤ 0.26

- 2(1-1/e) = 1.26 (**Pr**[rounded to 2] = 0.26)

Analysis of Warm-up Algorithm

- Three edge types
  - **Type 1:** only in M1, **Pr**[matched] ≥ 0.58
  - **Type 2:** only in M2, **Pr**[matched] ≥ 0.148
  - **Type 3:** in both M1 and M2, **Pr**[matched] ≥ 0.632
- Two cases for edges
  - **Small:** $f_e \leq 1/2$, **Pr**[Type 1] = **Pr**[Type 2] $= f_e$
  - **Large:** $f_e > 1/2$, **Pr**[Type 1] = **Pr**[Type 2] $= 1 - f_e$
    Pr[Type 3] $= 2f_e - 1 \leq 0.26$

Analysis of Warm-up Algorithm

- "Small" edges ($f_e \leq 1/2$) achieve ratio of 0.729
- "Large" edges ($f_e > 1/2$) achieve ratio of 0.688
  - Warm-up algorithm ratio: 0.688
  - Note: a vertex can have only one large neighbor

Combining two algorithms that favor small and large edges, respectively, gives our edge-weighted ratio: 0.705

Unweighted & Vertex-weighted

Unweighted Variant

- Previous best: $0.7293 = 1\text{-}2/e^2$ (Jaillet and Lu)
  - Their analysis was tight
- Negative result: $0.86 = 1\text{-}1/e^2$ (Manshadi *et al*)
- Our result: $0.7299 > 1\text{-}2/e^2$

"In some sense, the ratio of $1 - 2/e^2$ achieved in [56] for the integral case, is a nice 'round' number, and one may suspect that it is the correct answer."
- Aranyak Mehta, *Online Matching and Ad Allocation* (Open question 3)

Random Lists Algorithm (Jaillet and Lu)

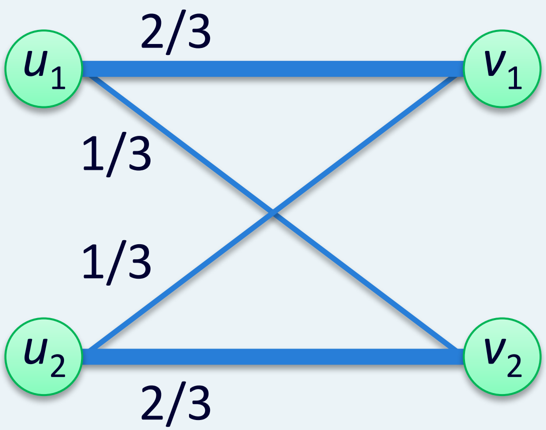
- Adaptive algorithm for online phase
- Requires special LP solution vector
  - Values in {0, 1/3, 2/3}
  - Certain short cycles removed
- Offline phase
  - LP edge constraint: $f_e \leq 2/3$ instead of $f_e \leq 1\text{-}1/e$
  - Cycle breaking step

Our contributions

- Use scaled dependent rounding
  - Multiply by 3, round, divide by 3, achieves values in {0, 1/3, 2/3} if all $f_e \leq 2/3$
- This allows for tighter LP constraints
  - Edge constraint: $f_e \leq 1\text{-}1/e$ instead of $f_e \leq 2/3$
  - Pair of edges constraint: $f_e + f_{e'} \leq 1\text{-}1/e^2$ for all e, e' in neighborhood of an offline vertex u
- Refined cycle breaking subroutine

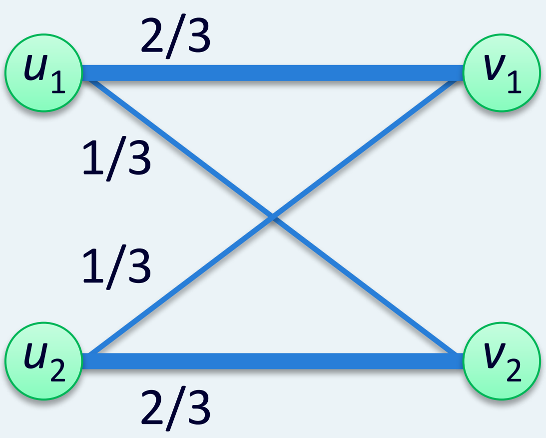
Bottleneck from Jaillet and Lu

- Length 4 cycles limit performance to $1-2/e^2$
- Some cycles can be broken, but not this one



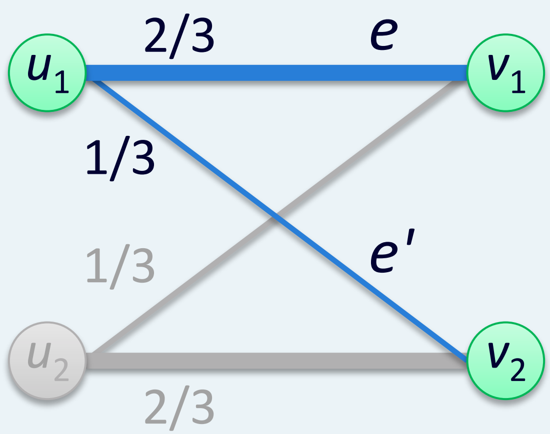
Comparison

- Jaillet and Lu: cycle occurs deterministically
- This work: cycle occurs with probability ≤ 0.89



Comparison

- Jaillet and Lu: cycle occurs deterministically
- This work: cycle occurs with probability ≤ 0.89



- Note: $f_e = 2/3$ and $f_e + f_{e'} = 1$
- We add constraints:
  - $f_e < 1\text{-}1/e = 0.63$
  - $f_e + f_{e'} < 1\text{-}1/e^2 = 0.86$
- Our bottleneck for this cycle:
  - $3(f_e) < 3(1\text{-}1/e) = 1.89$

Unweighted and Vertex-weighted

Scaled dependent rounding allows us to use tighter constraints and still produce a nice LP solution in {0, 1/3, 2/3}.

Stochastic Rewards

Stochastic Rewards

- Edges have (indep.) probabilities p_e of existing
  - Choose an edge to match, then find out if it exists
  - Motivation: pay-per-click advertising
- Generalization of the edge-weighted problem
- LP-based algorithm gets ratio of $1-1/e = 0.63$

For the natural benchmark LP, $1-1/e$ is tight!
Open: is there is a better LP or negative result?

Conclusion

- Edge-weighted
  - Scaled dependent rounding and balancing leads to improved results
- Unweighted/Vertex-Weighted
  - Scaled dependent rounding allows for tighter LP constraints to break the $1-2/e^2$ barrier
- Stochastic Rewards
  - Introduced problem, achieve ratio of $1-1/e$

Future Work

- Edge-weighted
  - Adaptive approach?
- Unweighted/Vertex-Weighted
  - Can we close the gap between 0.7299 and 0.86?
  - Use scaled dependent rounding with some $k > 3$?
- Stochastic Rewards
  - Tighter benchmark LP?
  - Negative result? Is beating 1-1/e even possible?
    - For related problem of non-integral arrival rates, no non-adaptive algorithm can beat 1-1/e (Manshadi et al)

Thank You!