Decomposition of Probability Marginals for Security Games in Abstract Networks (and Ideal Clutters)

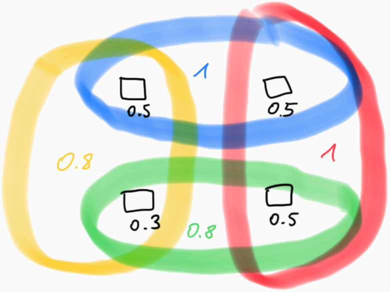
Jannik Matuschke

KU Leuven

From Marginals to Distributions

Input:

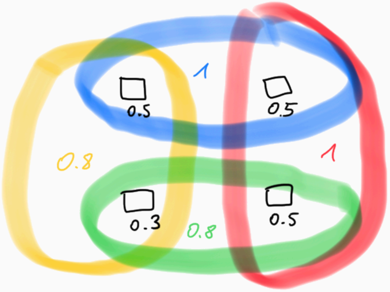
- ground set E
- set system $\mathcal{P} \subseteq 2^E$
- requirements $\pi \in [0,1]^{\mathcal{P}}$
- marginals $\rho \in [0,1]^E$



From Marginals to Distributions

Input:

- ground set E
- set system $\mathcal{P} \subseteq 2^E$
- requirements $\pi \in [0,1]^{\mathcal{P}}$
- marginals $\rho \in [0,1]^E$



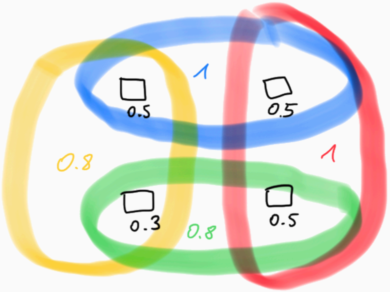
Goal: Find distribution for random set $S \subseteq E$ such that

$$\begin{aligned} \Pr\big[e \in S\big] &= \rho_e && \forall\, e \in E, \\ \Pr\big[P \cap S \neq \emptyset\big] &\geq \pi_P && \forall\, P \in \mathcal{P}. \end{aligned}$$

A Necessary Condition

All feasible marginals ρ must fulfil:

$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall P \in \mathcal{P} \quad (\star)$$

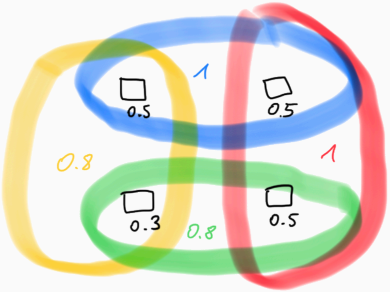


A Necessary Condition

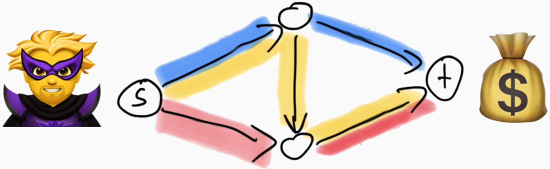
All feasible marginals ρ must fulfil:

$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

For which systems $(\mathcal{P}, \pi)$ is $(\star)$ also sufficent?



Motivation: A Security Game



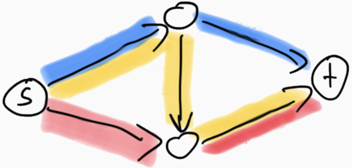
c_e: inspection cost

π_P: risk threshold

Inspect random set $S \subseteq E$:

$$
\begin{aligned}
\min \sum_{S \subseteq E} \sum_{e \in S} c_e \, x_S & \\
\text{s.t.} \sum_{S : P \cap S \neq \emptyset} x_S &\geq \pi_P \quad \forall\, P \in \mathcal{P} \\
\sum_{S \subseteq E} x_S &= 1 \\
x &\geq 0
\end{aligned}
$$

Motivation: A Security Game



c_e: inspection cost

π_P: risk threshold

Inspect random set $S \subseteq E$:

$$
\begin{aligned}
\min \quad & \sum_{S \subseteq E} \sum_{e \in S} c_e \, x_S \\
\text{s.t.} \quad & \sum_{S : P \cap S \neq \emptyset} x_S \geq \pi_P \quad \forall\, P \in \mathcal{P} \\
& \sum_{S \subseteq E} x_S = 1 \\
& x \geq 0
\end{aligned}
$$

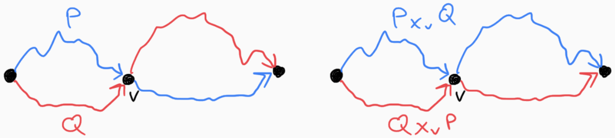
If $(\star)$ is sufficient:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e \, \rho_e \\
\text{s.t.} \quad & \sum_{e \in P} \rho_e \geq \pi_P \qquad \forall\, P \in \mathcal{P} \\
& \rho \in [0,1]^E
\end{aligned}
$$

$\mathcal{P} = \{s\text{-}t\text{-paths in a DAG}\}$, two settings for π:

(A) **Affine requirements:** $\pi_P = 1 - \sum_{e \in P} \mu_e$ for some $\mu \in [0,1]^E$

(C) **Conservation law:** $\pi_P + \pi_Q = \pi_{P \times_v Q} + \pi_{Q \times_v P}$ for $P, Q \in \mathcal{P}, v \in P \cap Q$



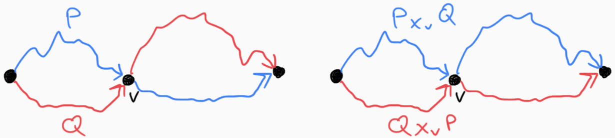
Previous Results

$\mathcal{P} = \{s\text{-}t\text{-paths in a DAG}\}$, two settings for π:

(A) **Affine requirements:** $\pi_P = 1 - \sum_{e \in P} \mu_e$ for some $\mu \in [0,1]^E$

(C) **Conservation law:** $\pi_P + \pi_Q = \pi_{P \times_v Q} + \pi_{Q \times_v P}$ for $P, Q \in \mathcal{P}, v \in P \cap Q$

Note: (A) $\Rightarrow$ (C).

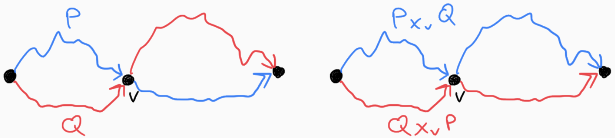


$\mathcal{P} = \{s\text{-}t\text{-paths in a DAG}\}$, two settings for π:

(A) Affine requirements: $\pi_P = 1 - \sum_{e \in P} \mu_e$ for some $\mu \in [0,1]^E$

(C) Conservation law: $\pi_P + \pi_Q = \pi_{P \times_v Q} + \pi_{Q \times_v P}$ for $P, Q \in \mathcal{P}, v \in P \cap Q$

Note: (A) $\Rightarrow$ (C).



Their results:

- For (C): $(\star)$ is sufficient.
- For (A): Decomposition can be computed efficiently.
- Consequence: Computation of Nash equilibria for security game on DAG.

New Results

	DAGs (Dahan et al.)
Affine	efficient algorithm
Conservation	$(\star)$ sufficient (exp.-time algorithm)

New Results

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)
Affine	efficient algorithm	efficient algorithm (explicit description) $\oplus$
Conservation	$(\star)$ sufficient (exp.-time algorithm)	

$\oplus$ combinatorial shortest-path algorithm for abstract networks

New Results

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)
Affine	efficient algorithm	efficient algorithm (explicit description) ⊕
	↑	
Conservation	$(\star)$ sufficient (exp.-time algorithm)	

⊕ combinatorial shortest-path algorithm for abstract networks

New Results

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)
Affine	efficient algorithm ↑	efficient algorithm (explicit description) $\oplus$
Conservation	$(\star)$ sufficient (exp.-time algorithm)	

$\oplus$ combinatorial shortest-path algorithm for abstract networks

Also: NP-hard to decide feasibility of given ρ in general systems

New Results

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)	Max-Flow/Min-Cut (aka. ideal clutters)
Affine	efficient algorithm ↑	efficient algorithm (explicit description) ⊕	characterize (⋆)-sufficiency (oracle-poly)
Conservation	(⋆) sufficient (exp.-time algorithm)		

⊕ combinatorial shortest-path algorithm for abstract networks

Also: NP-hard to decide feasibility of given ρ in general systems

New Results

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)	Max-Flow/Min-Cut (aka. ideal clutters)
Affine	efficient algorithm ↑	efficient algorithm (explicit description) ⊕	characterize ($\star$)-sufficiency (oracle-poly)
Conservation	($\star$) sufficient (exp.-time algorithm)	($\star$) sufficient (oracle-poly)	

⊕ combinatorial shortest-path algorithm for abstract networks

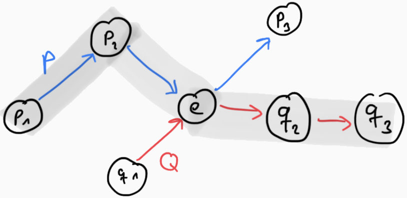
Also: NP-hard to decide feasibility of given ρ in general systems

Feasible Decompositions in Abstract Networks

Abstract network:

- set system $(E, \mathcal{P})$
- order $\preceq_P$ for every $P \in \mathcal{P}$
- for every $P, Q \in \mathcal{P}$ and $e \in P \cap Q$:

$$P \times_e Q \in \mathcal{P} \quad \text{contained in} \quad \{p \in P \; : \; p \preceq_P e\} \cup \{q \in Q \; : \; e \preceq_Q q\}.$$



$$P \times_e Q = q_3 \to p_2 \to q_2$$

Feasible Decompositions in Abstract Networks (when $\pi_P = 1 - \sum_{e \in P} \mu_e$)

$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

Construct random $S \subseteq E$ with

$$\Pr\Big[e \in S\Big] \;=\; \rho_e \qquad \forall\, e \in E,$$

$$\Pr\Big[P \cap S \neq \emptyset\Big] \;\geq\; \pi_P \qquad \forall\, P \in \mathcal{P}.$$

Feasible Decompositions in Abstract Networks

(when $\pi_P = 1 - \sum_{e \in P} \mu_e$)

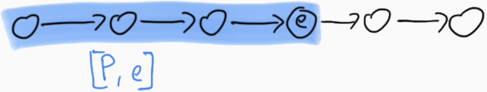
$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

Construct random $S \subseteq E$ with

$$\Pr\Big[e \in S\Big] = \rho_e \qquad \forall\, e \in E,$$

$$\Pr\Big[P \cap S \neq \emptyset\Big] \geq \pi_P \qquad \forall\, P \in \mathcal{P}.$$

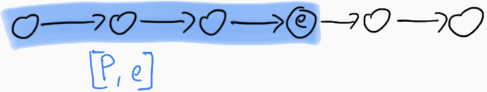
$$\alpha_e := \min_{P \in \mathcal{P}} \sum_{f \in [P,e]} \rho_f + \mu_f$$



Feasible Decompositions in Abstract Networks (when $\pi_P = 1 - \sum_{e \in P} \mu_e$)

$$S_\tau := \left\{ e \in E \,:\, \alpha_e - \rho_e \le \tau \le \alpha_e \right\}$$

with $\alpha_e := \min_{P \in \mathcal{P}} \sum_{f \in [P,e]} \rho_f + \mu_f$ and $\tau \sim U[0,1]$



Theorem. S_τ is a feasible decomposition of ρ.

$$\sum_{e \in P} \rho_e \ge \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

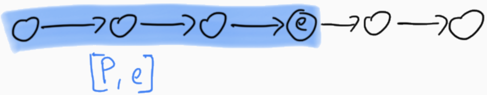
Construct random $S \subseteq E$ with

$$\Pr\Big[e \in S\Big] = \rho_e \qquad \forall\, e \in E,$$

$$\Pr\Big[P \cap S \neq \emptyset\Big] \ge \pi_P \qquad \forall\, P \in \mathcal{P}.$$

$$S_\tau := \Big\{ e \in E \,:\, \alpha_e - \rho_e \leq \tau \leq \alpha_e \Big\}$$

with $\alpha_e := \min_{P \in \mathcal{P}} \sum_{f \in [P,e]} \rho_f + \mu_f$ and $\tau \sim U[0,1]$



$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall P \in \mathcal{P} \quad (\star)$$

Construct random $S \subseteq E$ with

$$\Pr\Big[e \in S\Big] = \rho_e \qquad \forall e \in E,$$

$$\Pr\Big[P \cap S \neq \emptyset\Big] \geq \pi_P \qquad \forall P \in \mathcal{P}.$$

Theorem. S_τ is a feasible decomposition of ρ.

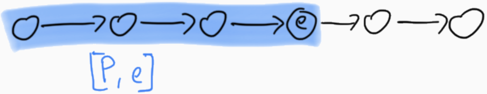
Proof sketch. Want to show: $\Pr\left[S_\tau \cap P \neq \emptyset\right] \geq 1 - \sum_{e \in P} \mu_e$

Feasible Decompositions in Abstract Networks

(when $\pi_P = 1 - \sum_{e \in P} \mu_e$)

$$S_\tau := \left\{ e \in E \,:\, \alpha_e - \rho_e \le \tau \le \alpha_e \right\}$$

with $\alpha_e := \min_{P \in \mathcal{P}} \sum_{f \in [P,e]} \rho_f + \mu_f$ and $\tau \sim U[0,1]$



$$\sum_{e \in P} \rho_e \ge \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

Construct random $S \subseteq E$ with

$$\Pr\left[e \in S\right] = \rho_e \qquad \forall\, e \in E,$$

$$\Pr\left[P \cap S \neq \emptyset\right] \ge \pi_P \qquad \forall\, P \in \mathcal{P}.$$

Theorem. S_τ is a feasible decomposition of ρ.

Proof sketch. Want to show: $\Pr\left[S_\tau \cap P \neq \emptyset\right] \ge 1 - \sum_{e \in P} \mu_e$

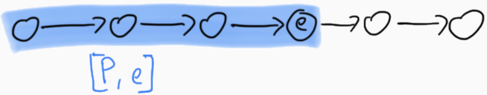
By induction: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \le \alpha_e\right] \ge \alpha_e - \sum_{f \in [P,e]} \mu_f$

Feasible Decompositions in Abstract Networks

(when $\pi_P = 1 - \sum_{e \in P} \mu_e$)

$$S_\tau := \left\{ e \in E \,:\, \alpha_e - \rho_e \leq \tau \leq \alpha_e \right\}$$

with $\alpha_e := \min_{P \in \mathcal{P}} \sum_{f \in [P,e]} \rho_f + \mu_f$ and $\tau \sim U[0,1]$



$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

Construct random $S \subseteq E$ with

$$\Pr\left[e \in S\right] = \rho_e \qquad \forall\, e \in E,$$

$$\Pr\left[P \cap S \neq \emptyset\right] \geq \pi_P \qquad \forall\, P \in \mathcal{P}.$$

Theorem. S_τ is a feasible decomposition of ρ.

Proof sketch.

Want to show: $\Pr\left[S_\tau \cap P \neq \emptyset\right] \geq 1 - \sum_{e \in P} \mu_e$

By induction: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \leq \alpha_e\right] \geq \alpha_e - \sum_{f \in [P,e]} \mu_f$

By $(\star)$: $\alpha_t \geq 1$ for last element t of P

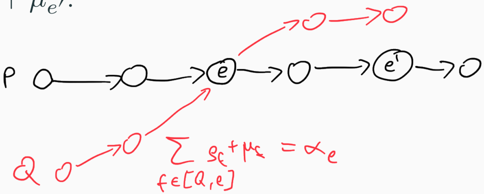
Induction Step

Hypothesis: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \;\wedge\; \tau \leq \alpha_e\right] \;\geq\; \alpha_e - \sum_{f \in [P,e]} \mu_f$

Induction Step

Hypothesis: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \leq \alpha_e\right] \geq \alpha_e - \sum_{f \in [P,e]} \mu_f$

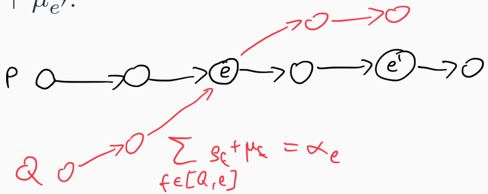
Claim: There is $e' \succ_P e$ with $\alpha_{e'} \leq \alpha_e + \rho_{e'} + \mu_{e'}$.



Induction Step

Hypothesis: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \leq \alpha_e\right] \geq \alpha_e - \sum_{f \in [P,e]} \mu_f$

Claim: There is $e' \succ_P e$ with $\alpha_{e'} \leq \alpha_e + \rho_{e'} + \mu_{e'}$.

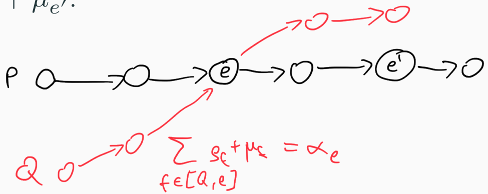


$e' :=$ first edge on $Q \times_e P$ not in $[Q,e]$

Induction Step

Hypothesis: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \leq \alpha_e\right] \geq \alpha_e - \sum_{f \in [P,e]} \mu_f$

Claim: There is $e' \succ_P e$ with $\alpha_{e'} \leq \alpha_e + \rho_{e'} + \mu_{e'}$.



$e' :=$ first edge on $Q \times_e P$ not in $[Q,e]$

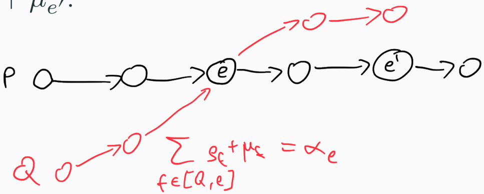
Induction step: Replace e by e'.

RHS increases by at most $\rho_{e'}$.

Induction Step

Hypothesis: $\Pr\left[S_\tau \cap [P,e] \neq \emptyset \wedge \tau \leq \alpha_e\right] \geq \alpha_e - \sum_{f\in[P,e]} \mu_f$

Claim: There is $e' \succ_P e$ with $\alpha_{e'} \leq \alpha_e + \rho_{e'} + \mu_{e'}$.

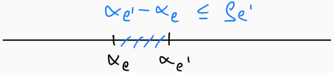


$e' :=$ first edge on $Q \times_e P$ not in $[Q, e]$

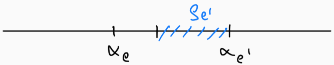
Induction step: Replace e by e'.

RHS increases by at most $\rho_{e'}$.

Case 1:



Case 2:



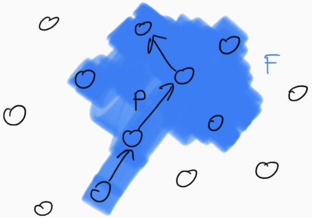
Shortest Paths in Abstract Networks

How Do We Access Abstract Networks?

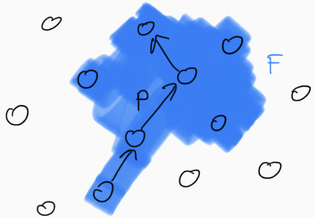
Membership oracle for an abstract network:

Given $F \subseteq E$, either

- return $P \in \mathcal{P}$ (and $\preceq_P$) with $P \subseteq F$,
- or assert that no such P exists.



How Do We Access Abstract Networks?



Membership oracle for an abstract network:

Given $F \subseteq E$, either

- return $P \in \mathcal{P}$ (and $\preceq_P$) with $P \subseteq F$,
- or assert that no such P exists.

McCormick (SODA 1996):

- Combinatorial algorithm for MAX FLOW in abstract networks using membership oracle (weakly poly-time).
- Strongly poly-time possible using stronger oracle? E.g., shortest-path oracle?

Shortest Paths in Abstract Networks

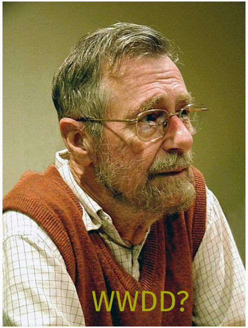
Given: abstract network $(E, \mathcal{P})$, costs $c \in \mathbb{R}_+^E$

Task: find $P \in \mathcal{P}$ minmizing $c(P) := \sum_{e \in P} c_e$

Shortest Paths in Abstract Networks

Given: abstract network $(E, \mathcal{P})$, costs $c \in \mathbb{R}_+^E$

Task: find $P \in \mathcal{P}$ minmizing $c(P) := \sum_{e \in P} c_e$

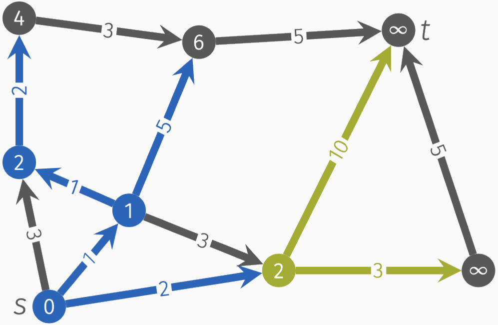


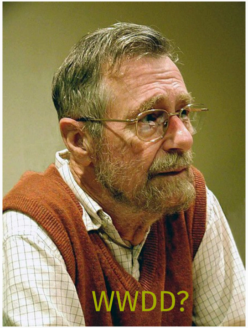
Shortest Paths in Abstract Networks

Given: abstract network $(E, \mathcal{P})$, costs $c \in \mathbb{R}_+^E$

Task: find $P \in \mathcal{P}$ minmizing $c(P) := \sum_{e \in P} c_e$

Dijkstra's Algorithm





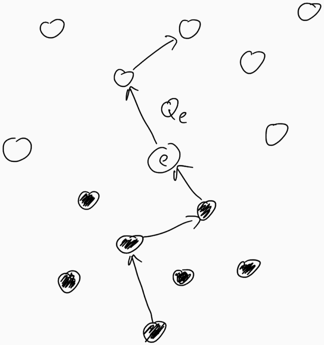
labels ϕ_e

paths Q_e

$\phi_e = \sum_{f \in [Q_e, e]} c_f$

Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

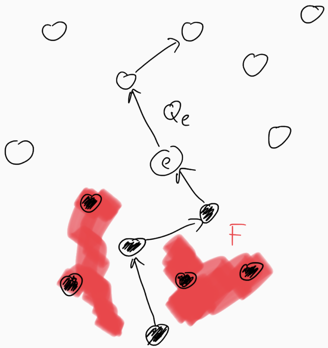


Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

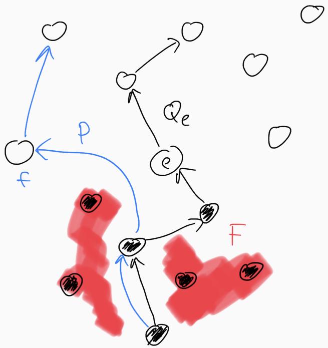
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

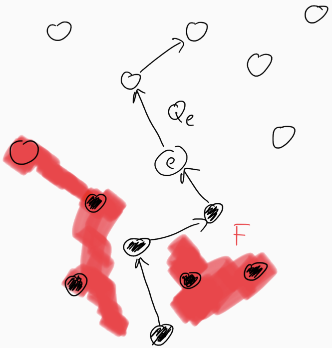
$\texttt{process}(e)$

- $F := T \backslash [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \backslash F$:

  $f := \min_{\preceq_P} P \backslash [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

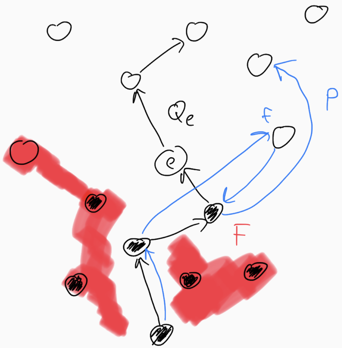
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

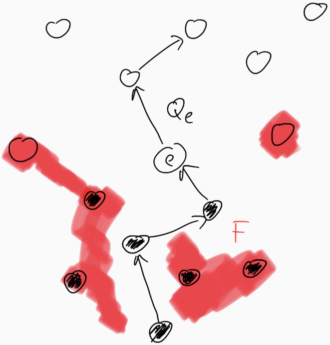
`process`(e)

- $F := T \backslash [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \backslash F$:

  $f := \min_{\preceq_P} P \backslash [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

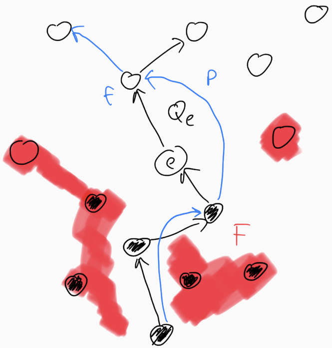
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f



Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

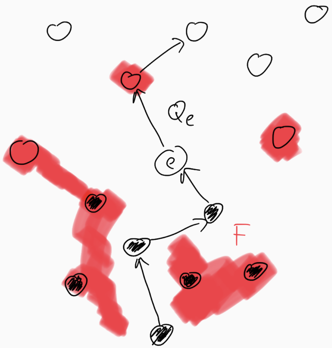
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f
- $T := T \cup \{e\}$



How to find all relevant ways to continue $[Q_e, e]$?

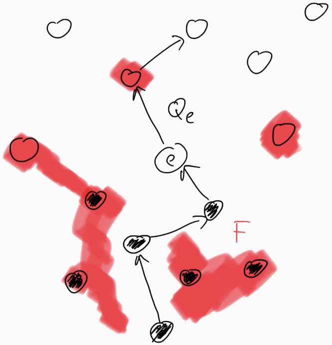
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f
- $T := T \cup \{e\}$



Lemma. After $\texttt{process}(e)$, for every $P \in \mathcal{P}$ with $e \in P$:

- there is $f \in P \setminus T$ with $\phi_f \leq \phi_e + c_f$.

How to find all relevant ways to continue $[Q_e, e]$?

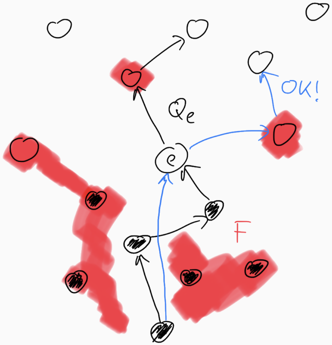
$\texttt{process}(e)$

- $F := T \backslash [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \backslash F$:

  $f := \min_{\preceq_P} P \backslash [Q_e, e]$

  $F := F \cup \{f\}$

  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f
- $T := T \cup \{e\}$



Lemma. After $\texttt{process}(e)$, for every $P \in \mathcal{P}$ with $e \in P$:

- there is $f \in P \backslash T$ with $\phi_f \leq \phi_e + c_f$.

Processing Elements

How to find all relevant ways to continue $[Q_e, e]$?

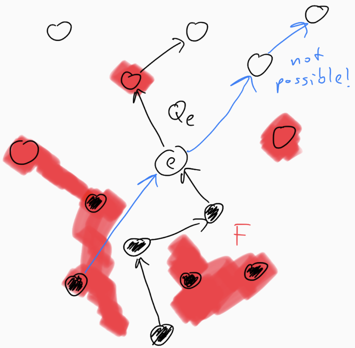
$\texttt{process}(e)$

- $F := T \setminus [Q_e, e]$
- while $\exists P \in \mathcal{P}$ with $P \subseteq E \setminus F$:

  $f := \min_{\preceq_P} P \setminus [Q_e, e]$

  $F := F \cup \{f\}$

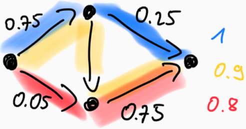
  if $c([P, f]) < \phi_f$ then update ϕ_f and Q_f
- $T := T \cup \{e\}$

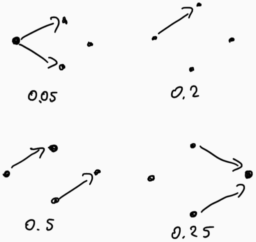


Lemma. After $\texttt{process}(e)$, for every $P \in \mathcal{P}$ with $e \in P$:

- there is $f \in P \setminus T$ with $\phi_f \leq \phi_e + c_f$.

Conclusion





- **(⋆)-sufficiency** allows formulating problems via their marginals:

$$\sum_{e \in P} \rho_e \geq \pi_P \quad \forall\, P \in \mathcal{P} \quad (\star)$$

- many systems are (⋆)-sufficient, including **abstract networks**
- feasible decompositions can be computed via a **shortest-path algorithm**

Overview & Open Questions

	DAGs (Dahan et al.)	Abstract Networks (incl. digraphs w. cycles)	Max-Flow/Min-Cut
Affine	efficient algorithm	efficient algorithm (explicit description) $\oplus$	characterize $(\star)$-sufficiency
	$\Updownarrow$		
Conservation	$(\star)$-sufficient (exp.-time algorithm)	$(\star)$-sufficient (oracle-poly)	TDI systems?

$\oplus$ combinatorial shortest-path algorithm for abstract networks

Strongly poly-time algorithm for Abstract Max Flow?

Also: NP-hard to decide feasibility of given ρ in general systems

Poly-time algorithms for some non-$(\star)$-sufficient systems?

$(\star)$-sufficiency under additional constraints on decomposition?

Copyright Note