Two-stage assignment problem under uncertainty by message-passing

Alfredo Braunstein

Politecnico di Torino

Parma, June 2010

with Fabrizio Altarelli, Abolfazl Ramezanpour and Riccardo Zecchina

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Why stochastic matchings

- Stochastic optimization
  - Almost *all* real-world optimization problems involve uncertainty
  - Two-stage stochastic optimization is a simple model for it:
    1. The problem input is only partially known (probabilistically) in advance when part of the decisions have to be taken.
    2. The full input is revealed on a second stage, and the rest of the decisions can then be taken.
- In the literature
  - Extensely studied in the OR community when the optimization is a LP (but not only): "Stochastic Programming"
  - Multi-stage (online) matching: [RANKING, Karp, Vazirani & Vazirani, 1990]
  - "The simplest two-stage problem": The two-stage stochastic matching problem was introduced in [Katriel, Kenyon-Mathieu & Upfal, 2008]. Several variants have been proved APX-Hard.

Matchings: Notation and definitions

Given $G = (V, E)$, a **matching** is a subset $E' \subset E$ of pairwise disjoint edges.

- For every $e \in E$, we define $\mathbf{x} \in \{0, 1\}^E$ as

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

- E' is a matching iff $\sum_{j \in \partial i} x_{ij} \leq 1 \forall i \in V$.
- The size of the matching

$$-H_G(\mathbf{x}) = \sum_{ij \in E} x_{ij}$$

- Maximum cardinality matching is "easy" (e.g. Edmonds's "Paths, trees and flowers" algorithm)

Matchings: Notation and definitions

Given $G = (V, E)$, a **matching** is a subset $E' \subset E$ of pairwise disjoint edges.

- For every $e \in E$, we define $\mathbf{x} \in \{0, 1\}^E$ as

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

- E' is a matching iff $\sum_{j \in \partial i} x_{ij} \leq 1 \forall i \in V$.
- The size of the matching

$$-H_G(\mathbf{x}) = \sum_{ij \in E} x_{ij}$$

- Maximum cardinality matching is "easy" (e.g. Edmonds's "Paths, trees and flowers" algorithm)

Matchings: Notation and definitions

Given $G = (V, E)$, a **matching** is a subset $E' \subset E$ of pairwise disjoint edges.

- For every $e \in E$, we define $\mathbf{x} \in \{0,1\}^E$ as

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

- E' is a matching iff $\sum_{j \in \partial i} x_{ij} \leq 1 \forall i \in V$.
- The size of the matching

$$-H_G(\mathbf{x}) = \sum_{ij \in E} x_{ij}$$

- Maximum cardinality matching is "easy" (e.g. Edmonds's "Paths, trees and flowers" algorithm)

Matchings: Notation and definitions

Given $G = (V, E)$, a **matching** is a subset $E' \subset E$ of pairwise disjoint edges.

- For every $e \in E$, we define $\mathbf{x} \in \{0,1\}^E$ as

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

- E' is a matching iff $\sum_{j \in \partial i} x_{ij} \leq 1 \forall i \in V$.
- The size of the matching

$$-H_G(\mathbf{x}) = \sum_{ij \in E} x_{ij}$$

- Maximum cardinality matching is "easy" (e.g. Edmonds's "Paths, trees and flowers" algorithm)

Matchings: Notation and definitions

Given $G = (V, E)$, a **matching** is a subset $E' \subset E$ of pairwise disjoint edges.

- For every $e \in E$, we define $\mathbf{x} \in \{0,1\}^E$ as

$$x_e = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

- E' is a matching iff $\sum_{j \in \partial i} x_{ij} \leq 1 \forall i \in V$.
- The size of the matching

$$-H_G(\mathbf{x}) = \sum_{ij \in E} x_{ij}$$

- Maximum cardinality matching is "easy" (e.g. Edmonds's "Paths, trees and flowers" algorithm)

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1-p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1 - t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1-p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1 - t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1 - t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1-p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1 - t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

$$\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})$$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1-t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

$$\mathbb{E}_{\mathbf{t}} \min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})$$

Two-stage Stochastic Matching

- We are given a bipartite graph $G = (V = L \cup R, E \subset L \times R)$ with L sub-divided as $L = L_1 \cup L_2$.
- G is known, but each $l \in L_2$ will be present with independent probability p_l.
- **Decisions** are divided in two stages:
  1. We **choose** a matching $\mathbf{x}$ for nodes in L_1 to nodes in R knowing just G and $\mathbf{p}$
  2. *Then* a vector $\mathbf{t} \in \{0,1\}^{L_2}$ is extracted with $P(\mathbf{t}) = \prod_{l \in L_2} p_l^{t_l} (1 - p_l)^{1 - t_l}$
  3. *Then* we can **choose** a matching $\mathbf{y}$ for nodes in $\{t_l = 1\}$ to R

The scope is to minimize the final energy (in average)

$$H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = -\sum_{l \in L_1} \sum_{r \in \partial l} x_{lr} - \sum_{l \in L_2} \sum_{r \in \partial l} y_{lr}$$

subject to matching conditions for $\mathbf{x} \cup \mathbf{y}$

$$\arg\min_{\mathbf{x}} \mathbb{E}_{\mathbf{t}} \min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})$$

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})}^{B(\mathbf{x},\mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*

Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = \hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k \in \partial i \setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t}, \mathbf{u}) \propto \prod_{ij} \delta(u_{ij} - f_{MS}(\mathbf{u}, \mathbf{x}, t_i)) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x}, \mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x}, \mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y})}^{B(\mathbf{x},\mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*

Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y}) = \hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k\in\partial i\setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t},\mathbf{u}) \propto \prod_{ij} \delta(u_{ij} - f_{MS}(\mathbf{u},\mathbf{x},t_i)) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x},\mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x},\mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y})}^{B(\mathbf{x},\mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*
Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y}) = \hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k\in\partial i\setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t},\mathbf{u}) \propto \prod_{ij}\delta(u_{ij} - f_{MS}(\mathbf{u},\mathbf{x},t_i))\prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x},\mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x},\mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y})}^{B(\mathbf{x},\mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*
Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y}) = \hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k\in\partial i\setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t},\mathbf{u}) \propto \prod_{ij} \delta(u_{ij} - f_{MS}(\mathbf{u},\mathbf{x},t_i)) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x},\mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x},\mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})}^{B(\mathbf{x}, \mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*
Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = \hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k \in \partial i \setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t}, \mathbf{u}) \propto \prod_{ij} \delta(u_{ij} - f_{MS}(\mathbf{u}, \mathbf{x}, t_i)) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x}, \mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x}, \mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \quad \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y})}^{B(\mathbf{x}, \mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*
Idea: Switch to message-space and BP/MS to factorize:

- MS: $\min_{\mathbf{y}} H(\mathbf{x}, \mathbf{t}, \mathbf{y}) = \hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k \in \partial i \setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x}, \mathbf{t}, \mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t}, \mathbf{u}) \propto \prod_{ij} \delta\left(u_{ij} - f_{MS}(\mathbf{u}, \mathbf{x}, t_i)\right) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x}, \mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x}, \mathbf{M})$ subject to BP constraints

Optimization by MS/BP (1)

- $$\min_{\mathbf{x}} \underbrace{\mathbb{E}_{\mathbf{t}} \overbrace{\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y})}^{B(\mathbf{x},\mathbf{t})}}_{A(\mathbf{x})}$$

Problem: neither A nor B is G-factorized *nicely*
Idea: Switch to message-space and BP/MS to factorize:

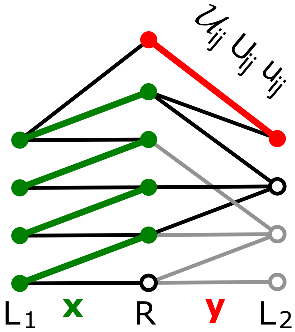
- MS: $\min_{\mathbf{y}} H(\mathbf{x},\mathbf{t},\mathbf{y}) = \hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ where $\mathbf{u}$ satisfy (local) MS equations, $u_{ij} = f_{MS}\left(\{u_{ki}\}_{k\in\partial i\setminus j}\right)$ for all $ij \in E$
- Use BP to compute the average $\mathcal{H}$ of the (G-factorized) observable $\hat{H}(\mathbf{x},\mathbf{t},\mathbf{u})$ in the probability space $P_{\mathbf{x}}(\mathbf{t},\mathbf{u}) \propto \prod_{ij} \delta(u_{ij} - f_{MS}(\mathbf{u},\mathbf{x},t_i)) \prod_i P_i(t_i)$.
  - BP messages are of the form $M_{ij}(u_{ij})$ and both BP equations and its expression for $\mathcal{H}(\mathbf{x},\mathbf{M})$ are again G-factorized.
- Employ MS again to minimize $\mathcal{H}(\mathbf{x},\mathbf{M})$ subject to BP constraints

Otimization by MS/BP (2)

- MS: $u_{ij} \in \{-1, 0, 1\}$,

$$u_{ij} = -\max\left\{-1, \max_{k \in \partial i \setminus j} u_{ki}\right\}$$

  - For $\langle c \rangle < e$, actually $u_{ij} \in \{-1, 1\}$ [Mézard & Zdeborova 2006]



- BP(MS): Parametrizing $U_{ij} = M_{ij}(u_{ij} = 1)$

$$U_{ij} = p_i \prod_{k \in i \setminus j} (1 - U_{ki}), \text{ where } p_i = 1 \text{ if } i \notin L_2$$

- MS(BP(MS)): For $\mathbf{U}_{ij} = \{U_{ki}\}_{k \in \partial i \setminus j}$

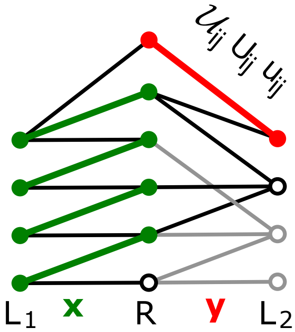
$$\mathcal{U}_{ij}(U_{ij}) = \max_{\mathbf{U}_{ij} : \mathbb{1}_{BP}(\mathbf{U}_{ij}, U_{ij}) = 1} \left\{ -\mathcal{H}_{ij}(U_{ij}, \mathbf{U}_{ij}) + \sum_{k \in \partial i \setminus j} \mathcal{U}_{ki}(U_{ki}) \right\}$$

Otimization by MS/BP (2)

- MS: $u_{ij} \in \{-1, 0, 1\}$,

$$u_{ij} = -\max\left\{-1, \max_{k \in \partial i \setminus j} u_{ki}\right\}$$

  - For $\langle c \rangle < e$, actually $u_{ij} \in \{-1, 1\}$ [Mézard & Zdeborova 2006]



- BP(MS): Parametrizing $U_{ij} = M_{ij}(u_{ij} = 1)$

$$U_{ij} = p_i \prod_{k \in i \setminus j} (1 - U_{ki}), \text{ where } p_i = 1 \text{ if } i \notin L_2$$

- MS(BP(MS)): For $\mathbf{U}_{ij} = \{U_{ki}\}_{k \in \partial i \setminus j}$

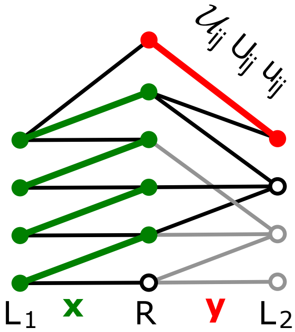
$$\mathcal{U}_{ij}(U_{ij}) = \max_{\mathbf{U}_{ij} : \mathbb{1}_{BP}(\mathbf{U}_{ij}, U_{ij}) = 1} \left\{ -\mathcal{H}_{ij}(U_{ij}, \mathbf{U}_{ij}) + \sum_{k \in \partial i \setminus j} \mathcal{U}_{ki}(U_{ki}) \right\}$$

Otimization by MS/BP (2)

- MS: $u_{ij} \in \{-1, 0, 1\}$,

$$u_{ij} = -\max\left\{-1, \max_{k \in \partial i \setminus j} u_{ki}\right\}$$

  - For $\langle c \rangle < e$, actually $u_{ij} \in \{-1, 1\}$ [Mézard & Zdeborova 2006]



- BP(MS): Parametrizing $U_{ij} = M_{ij}(u_{ij} = 1)$

$$U_{ij} = p_i \prod_{k \in i \setminus j} (1 - U_{ki}), \text{ where } p_i = 1 \text{ if } i \notin L_2$$

- MS(BP(MS)): For $\mathbf{U}_{ij} = \{U_{ki}\}_{k \in \partial i \setminus j}$

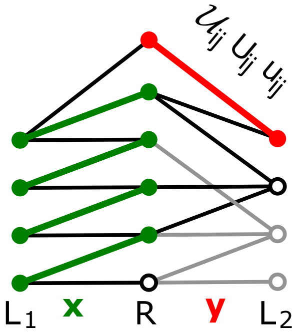
$$\mathcal{U}_{ij}(U_{ij}) = \max_{\mathbf{U}_{ij} : \mathbb{1}_{BP}(\mathbf{U}_{ij}, U_{ij}) = 1} \left\{ -\mathcal{H}_{ij}(U_{ij}, \mathbf{U}_{ij}) + \sum_{k \in \partial i \setminus j} \mathcal{U}_{ki}(U_{ki}) \right\}$$

Otimization by MS/BP (2)

- MS: $u_{ij} \in \{-1, 0, 1\}$,

$$u_{ij} = -\max\left\{-1, \max_{k \in \partial i \setminus j} u_{ki}\right\}$$

  - For $\langle c \rangle < e$, actually $u_{ij} \in \{-1, 1\}$ [Mézard & Zdeborova 2006]



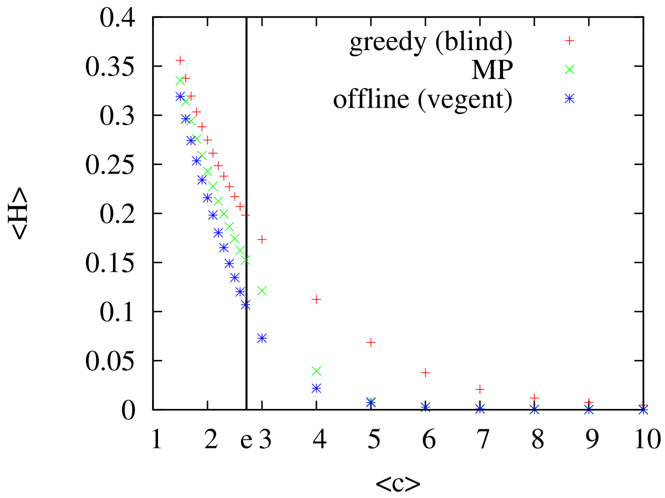
- BP(MS): Parametrizing $U_{ij} = M_{ij}(u_{ij} = 1)$

  $U_{ij} = p_i \prod_{k \in i \setminus j} (1 - U_{ki})$, where $p_i = 1$ if $i \notin L_2$

- MS(BP(MS)): For $\mathbf{U}_{ij} = \{U_{ki}\}_{k \in \partial i \setminus j}$

$$\mathcal{U}_{ij}(U_{ij}) = \max_{\mathbf{U}_{ij}: \mathbb{1}_{BP}(\mathbf{U}_{ij}, U_{ij}) = 1} \left\{ -\mathcal{H}_{ij}(U_{ij}, \mathbf{U}_{ij}) + \sum_{k \in \partial i \setminus j} \mathcal{U}_{ki}(U_{ki}) \right\}$$

Results



- $|L_1| = 1000, |L_2| = |R| = 2000$
- $p_l \in [0, 1]$ uniformly.
- Each point is an average of 130 to 350 instances

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x}_1} \max_{\mathbf{y}_1} \cdots \min_{\mathbf{x}_k} \max_{\mathbf{y}_k} H(\mathbf{x_1}, \ldots, \mathbf{x_k}, \mathbf{y_1}, \ldots, \mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x_1}} \max_{\mathbf{y_1}} \cdots \min_{\mathbf{x}_k} \max_{\mathbf{y}_k} H(\mathbf{x_1}, \ldots, \mathbf{x_k}, \mathbf{y_1}, \ldots, \mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x}_1}\max_{\mathbf{y}_1}\cdots\min_{\mathbf{x}_k}\max_{\mathbf{y}_k} H(\mathbf{x_1},\ldots,\mathbf{x_k},\mathbf{y_1},\ldots,\mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x_1}} \max_{\mathbf{y_1}} \cdots \min_{\mathbf{x}_k} \max_{\mathbf{y}_k} H(\mathbf{x_1}, \ldots, \mathbf{x_k}, \mathbf{y_1}, \ldots, \mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x_1}} \max_{\mathbf{y_1}} \cdots \min_{\mathbf{x}_k} \max_{\mathbf{y}_k} H(\mathbf{x_1}, \ldots, \mathbf{x_k}, \mathbf{y_1}, \ldots, \mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!

Open directions

- In-depth study for $\langle c \rangle > e$ and the transition at e
- Other problems: Stochastic Steiner, maximum independent set, stochastic sat
- Applications: kidney exchange, internet advertising (some in Fabrizio's talk)
- multi-stage optimization: games (e.g. zero-sum games: $\arg\min_{\mathbf{x_1}} \max_{\mathbf{y_1}} \cdots \min_{\mathbf{x}_k} \max_{\mathbf{y}_k} H(\mathbf{x_1}, \ldots, \mathbf{x_k}, \mathbf{y_1}, \ldots, \mathbf{y_k})$)
- Smarter parametrizations of nested distributions

Thank you!