Distributed Multiplicative Weights Methods for DCOP
(分散制約最適化問題に対する分散乗算型重み更新法)

ERATO感謝祭 Season II

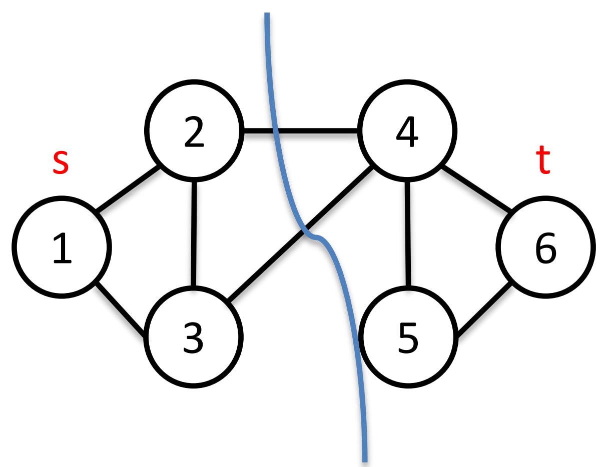
吉田 悠一
Joint with 波多野 大督

COP: 制約最適化問題

最小s-tカット

入力：グラフG = (V = {1, 2, ..., n}, E), 頂点s, t ∈ V

目的：s, t間の最小カットを求める



制約最適化問題

<u>制約最適化問題</u> (Constraint Optimization Problem, COP)

入力：グラフ$G = (V, E)$,

定義域D_i $(i \in V)$,

コスト関数$f_{ij}: D_i \times D_j \to \mathbb{R}_+$ $(ij \in E)$

目的：コストの低い割当を求める

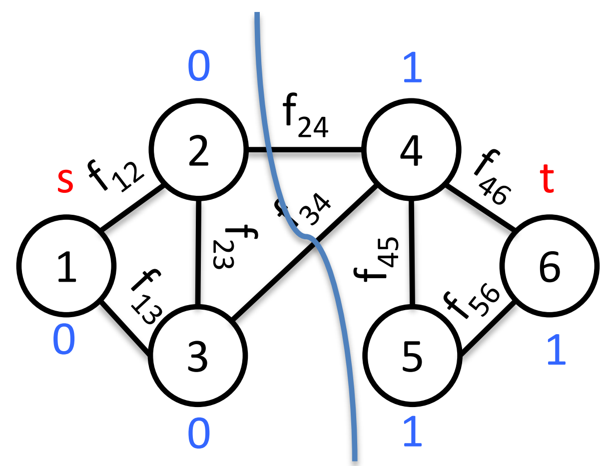
割当$\boldsymbol{\phi}$のコスト$f(\boldsymbol{\phi}) := \Sigma_{ij \in E} f_{ij}(\phi_i, \phi_j)$

<u>最小s-tカット</u>

$D_s = \{0\}$, $D_t = \{1\}$

$D_i = \{0, 1\}$ $(i \in V$ ¥ $\{s, t\})$

$f_{ij}(a, b) = [a \neq b]$



分散制約最適化問題

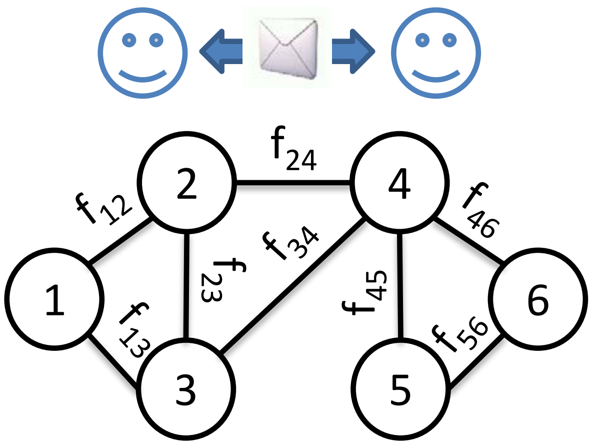
<u>分散制約最適化問題</u> (Distributed COP, DCOP)

入力：COPの入力(G, {D_i}, {f_{ij}})

- 各変数を管理するエージェントが存在
- 隣接するエージェントとのみ通信可能

目的：エージェントが協力してコストの低い割当を求める

- 通信量を抑える
- 計算時間を抑える



分散制約最適化問題の重要性

- 様々なマルチエージェントシステムをモデル化可能
  - エージェントが協調して動作
  - 会議日程調整問題
- 分散最適化が必要な状況
  - 入力が巨大で一台の計算機に載らない
  - 計算機等が物理的に分散している
  - 共有したくない情報がある（プライバシー）

本研究の貢献

分散最適化の課題

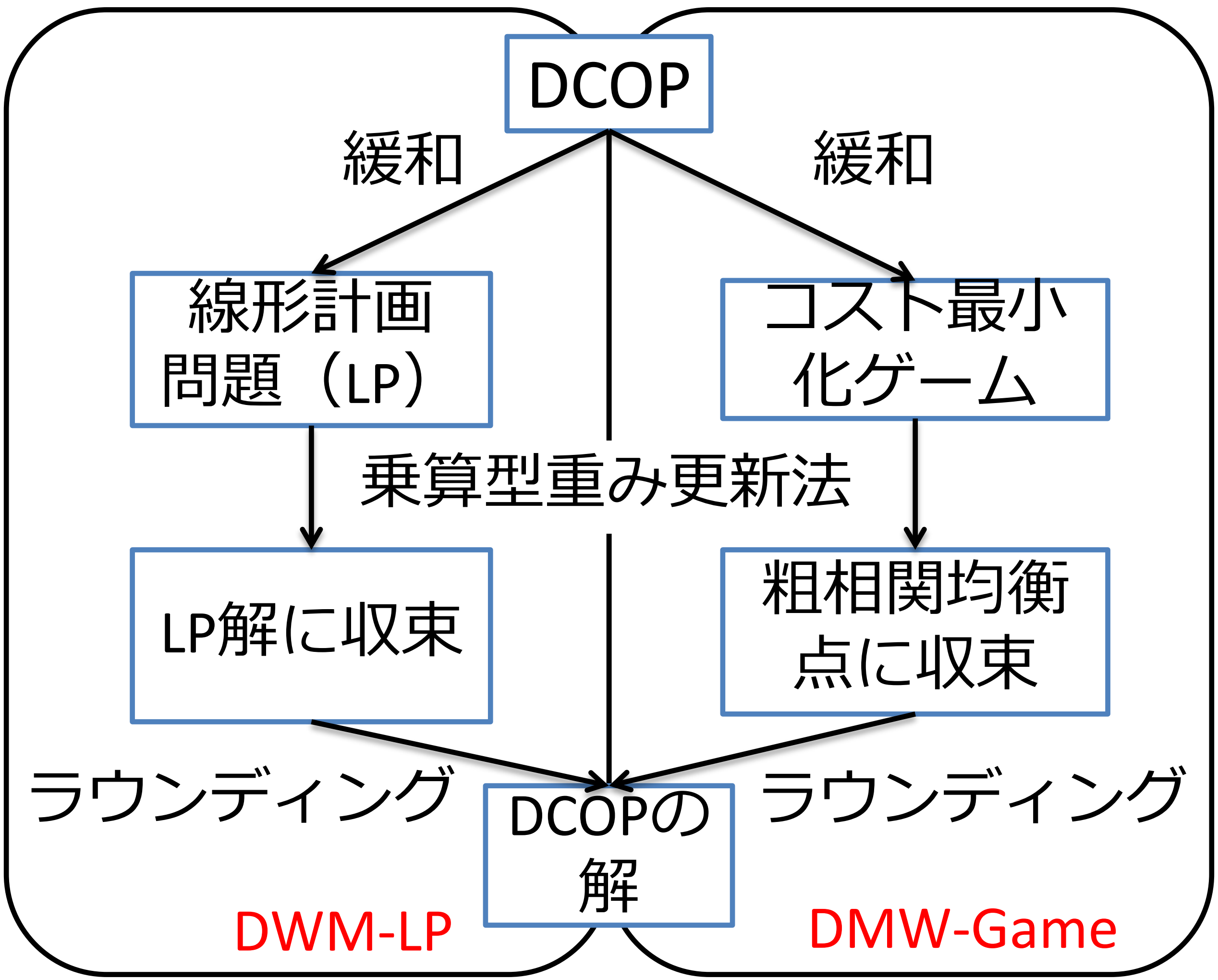
解の質 ⟷ 通信量・計算時間

乗算型重み更新法に基づいたアルゴリズムを二つ提案

- 局所的な通信のみ利用
- 実データ・人工データで、既存手法に対して
  - 最大2.75倍良い解が
  - より高速に

  得られた

提案アルゴリズムの概要



後悔最小化

<u>後悔（リグレット）最小化問題</u>

例：天気予測

- 複数の天気予報士が毎日天気予報する
- 次の日の天気から、各予報士の当たり外れが分かる
- 最良の予報士と同程度天気を当てたい

<u>乗算型重み更新法</u>：後悔最小化を達成する手法

- 予報士に信頼度（重み）を付与
- 信頼度に応じた確率で予報士を選んで予報を真似る
- 得られた結果から信頼度を更新する

乗算型重み更新法

乗算型重み更新法 (Multiplicative weight update method, MW)

入力：d個の戦略(＝予報士), 整数T(＝予報日数), 実数η

出力：{1, …., d}上の確率分布$\mathbf{p}^1, \ldots, \mathbf{p}^T$

（$\mathbf{p}^t$ = t日目の予報士を選ぶ戦略）

$\mathbf{w}^1 = (1, \ldots, 1)$

For t = 1 to T do

確率$p^t_a := w^t_a / \square \mathbf{w}^t \square_1$で戦略aを選択

コスト$\mathbf{c}^t = (c^t_1, \ldots, c^t_d)$を観測

$w^{t+1}_a := w^t_a(1 - \eta c^t_a)$ $(a \in \{1,\ldots,d\})$

乗算型重み更新法の理論保証

[定理] T = 1/ε², η = 1/√Tの時、任意の戦略aについて以下が成り立つ。

$$\frac{1}{T}\left(\sum_t \left\langle \boldsymbol{c}^t, \boldsymbol{p}^t \right\rangle - \sum_t c_a^t\right) \leq O(\varepsilon)$$

MWに従って予報士を選んだ時のコスト

予報士aを選択し続けた時のコスト

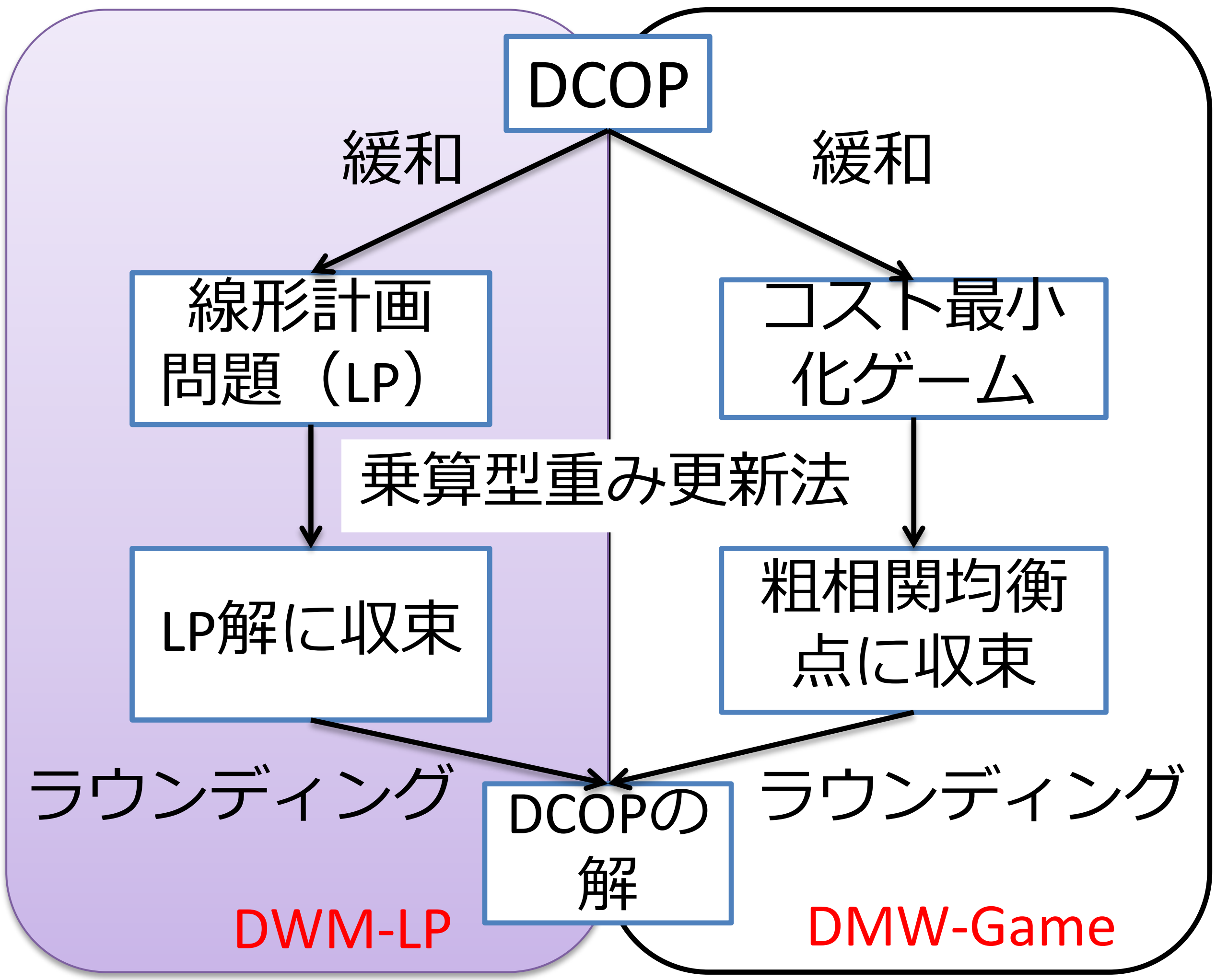
D_iのサイズは定数と仮定

[系] Tを十分大きく取れば、最良の戦略aに対するコストの差を任意に小さく出来る。

提案アルゴリズムのアイデア

- エージェントiは$f_i := \Sigma_{ij \in E} f_{ij}$を最小化したい
- エージェントiは戦略の分布μ_iを保持
  - 各隣接エージェントjから戦略分布μ_jを受け取る
  - 自分が戦略$a \in D_i$を選択した時のコストを計算
  - MWによって、自分の戦略分布μ_iを更新
- コストの定義により解の収束先が変わる

提案アルゴリズムの概要



DMW-LP: LP緩和

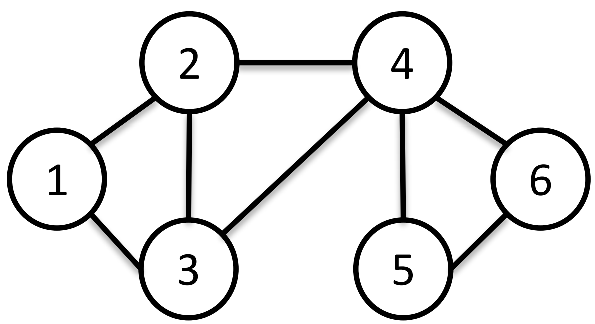
COPを線形計画問題 (LP) に緩和

min $\quad f(\{\mu_i\}) := \sum_{ij \in E} \mathop{E}_{(a,b)\sim\mu_{ij}} [f_{ij}(a,b)]$

s.t. (i) μ_iはD_i上の確率分布

(ii) μ_{ij} は $D_i \times D_j$上の確率分布

(iii) μ_{ij}のD_iにおける周辺分布はμ_iに一致



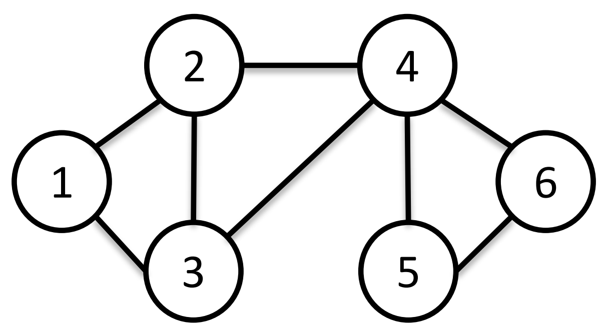
DMW-LP: コスト計算

エージェントiの各ステップでの動作

各隣接エージェントjから分布μ_jを受け取る

$\{\mu_j\}$を固定し、f_iをμ_iの関数とみた時の(劣)勾配を計算

勾配をコストとしてMWによりμ_iを更新する



DMW-LP: 理論保証

[定理]
$\{\mu_i\}$: DMW-LPがT = 1/ε²ステップ後に出力する分布
$\{\mu^*_i\}$: LP解
以下が成り立つ。

$$f(\{\mu_i\}) = f(\{\mu_i^*\}) + O(\varepsilon n)$$

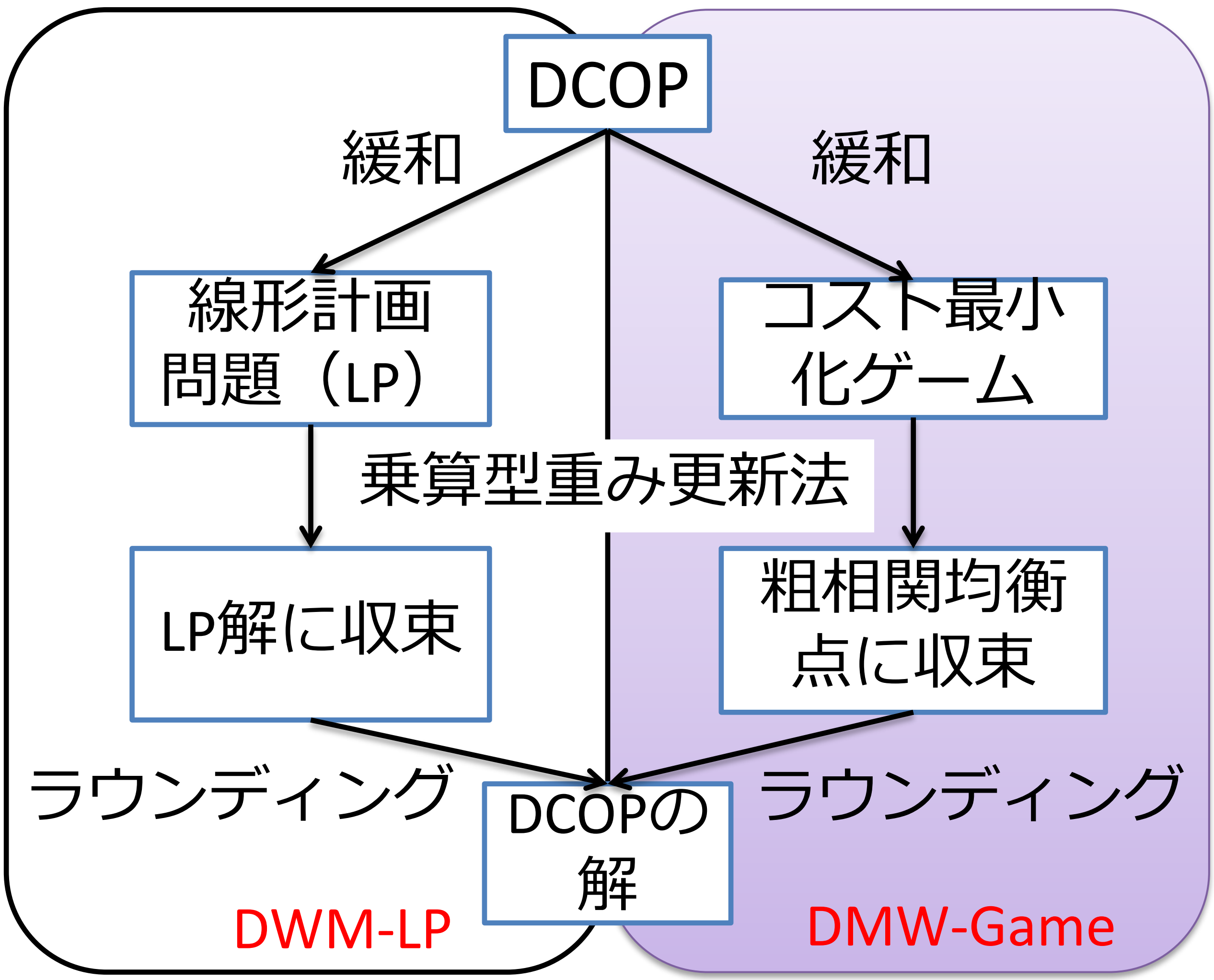
DMW-LPから得られる解

LP解

D_iのサイズは定数と仮定

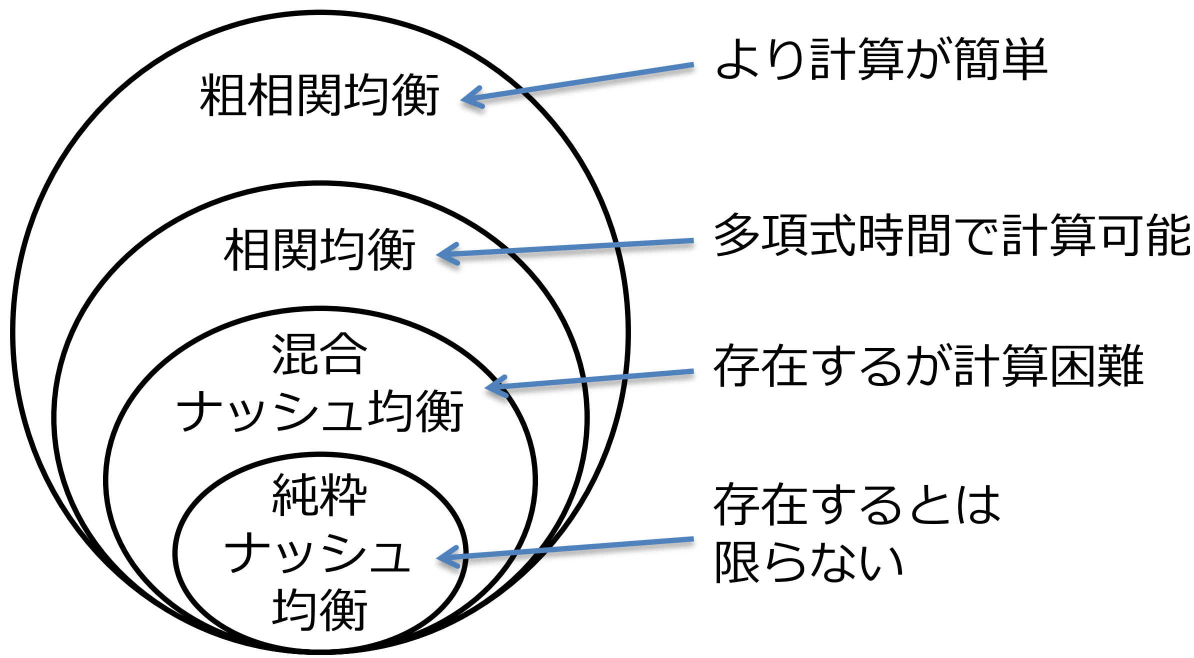
[系] Tが十分大きいと、LP解との差が任意に小さくなる

提案アルゴリズムの概要



DMW-Game: コスト最小化ゲーム

- エージェントiは$f_i := \Sigma_{ij \in E} f_{ij}$を最小化したい
- 各エージェントは周りを見て自分の戦略を変化させる
- 均衡点は良い解に違いない ⇒ どの均衡点に注目する？
- 粗相関均衡！



DMW-Game: コスト計算

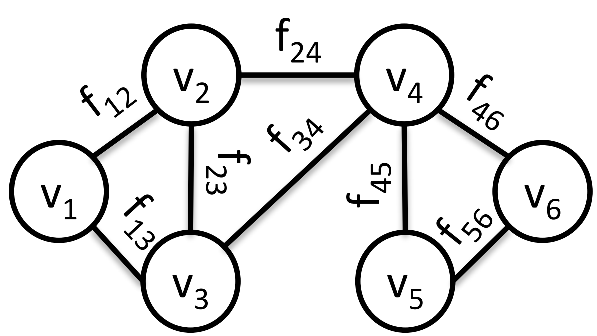
エージェントiの各ステップでの動作

各隣接エージェントjから分布μ_jを受け取る

$\{\mu_j\}$を固定した時の、戦略$a \in D_i$のコストc_aを計算

$\{c_a\}$をコストとしてMWによりμ_iを更新

出力$\mu = (1/T)\Sigma_t \mu^t_1 \times \cdots \times \mu^t_n$ (μ^t_i: tステップ目の分布μ_i)



DMW-Game: 理論保証

μ: DMW-GameがT = 1/ε²ステップ後に出力する分布
任意のi ∈ Vとc ∈ D_iについて以下が成り立つ。

$$\mathop{E}_{\boldsymbol{a}\sim\mu}\left[f(\boldsymbol{a})\right] = \mathop{E}_{\boldsymbol{a}\sim\mu}\left[f(a_1,\ldots,a_{i-1},c,a_{i+1},\ldots,a_n)\right] + O(\varepsilon n)$$

DMW-Gameから
得られる解

粗相関均衡

D_iのサイズは定数と仮定

Tが十分大きいと、粗相関均衡との差が任意に小さくなる

ラウンディング

確率分布μを割当φに丸める

DMW-LP

- 多数決: 一番確率の大きい要素を使う
- Distributed Stochastic Algorithm (DSA) 風
  - 変数の値を少しずつ決める
  - 既に値の決まっている隣接変数で条件付けした分布からサンプル

DMW-Game

- 多数決
- 多数決＋リスタート
  - 早く収束した変数を固定し、それ以外の変数の重みを初期化して再度実行

実験設定

計算環境: Ubuntu, Intel Core-i7 3770@3.4GHz, メモリ4GB

比較対象

- Maxsum (Messeage-Passing)
- DeQEDm, DeQEDa (Lagrange Duality)
- DSA
- MGM (Graphical-Game)

評価項目

- 解の質 = 得られた解のコスト / LPにより得られた下界
- 模擬実行時間

実験結果

Table 1: Average solution quality and simulated runtime (in msec) for (a) 100-node random networks with density 0.1, (b) 100-node scale-free networks, and (c) 100-node meeting scheduling problems.

Prob.	Cyc.	LP+Maj sol.	LP+Maj time	LP+DSA sol.	LP+DSA time	Game+Maj sol.	Game+Maj time	Game+Res sol.	Game+Res time	MaxSum sol.	MaxSum time	DeQED$_a$ sol.	DeQED$_a$ time	DeQED$_m$ sol.	DeQED$_m$ time	DSA sol.	DSA time	MGM sol.	MGM time
(a)	100	1.339	12.4	1.304	13.5	1.199	0.5	**1.184**	**0.3**	1.331	208.1	1.233	4.6	1.409	0.5	1.196	22.5	1.208	26.2
	200	1.330	24.5	1.301	27.4	1.188	1.2	**1.182**	**0.5**	1.331	272.3	1.217	9.3	1.363	1.0	1.198	29.9	1.197	33.7
	300	1.326	37.9	1.305	41.8	1.185	1.6	**1.182**	**0.7**	1.327	368.8	1.214	14.0	1.340	1.4	1.207	43.0	1.207	43.0
	400	1.328	49.8	1.305	56.2	1.183	1.9	**1.181**	**0.9**	1.317	454.8	1.214	18.7	1.327	1.9	1.203	42.7	1.201	51.5
	500	1.325	62.3	1.311	70.4	1.183	2.3	**1.181**	**1.1**	1.329	588.6	1.214	23.4	1.322	2.4	1.198	46.3	1.202	63.6
(b)	100	1.319	12.1	1.209	16.8	1.113	0.4	**1.094**	**0.3**	1.266	120.9	1.135	8.7	1.189	0.6	1.158	26.6	1.153	38.7
	200	1.286	24.4	1.185	32.4	1.099	0.8	**1.090**	**0.5**	1.296	213.4	1.128	17.7	1.156	1.2	1.156	33.7	1.153	51.3
	300	1.257	38.2	1.168	48.5	1.097	1.2	**1.089**	**0.8**	1.248	268.3	1.126	26.4	1.133	1.8	1.168	42.0	1.161	62.2
	400	1.251	49.8	1.167	64.6	1.095	1.6	**1.089**	**1.1**	1.217	309.7	1.126	35.3	1.138	2.4	1.157	48.7	1.153	76.3
	500	1.234	61.4	1.149	80.3	1.092	2.0	**1.089**	**1.3**	1.220	396.5	1.126	44.1	1.131	3.0	1.149	58.1	1.142	86.6
(c)	100	2001	12.7	1983	12.9	970	0.4	**914**	**0.2**	3081	138.1	2402	2.2	5278	0.7	1011	43.1	1135	54.7
	200	1706	23.4	1838	24.5	925	0.8	**914**	**0.4**	2829	288.5	1591	5.1	1970	1.4	1143	58.8	1065	81.0
	300	1589	33.3	1793	36.0	925	1.3	**914**	**0.6**	2520	364.3	1499	7.7	1652	2.1	1107	74.7	1099	105.3
	400	1614	43.8	1702	48.2	925	1.7	**914**	**0.8**	2648	450.7	1469	10.3	1583	2.8	996	89.8	1096	128.0
	500	1678	54.2	1810	60.4	940	2.1	**914**	**0.9**	2516	551.2	1469	13.4	1548	3.6	1094	106.0	1144	152.5

数msで計算可能！

まとめ

乗算型重み更新法に基づく解法を二つ提案

- DMW-LP: 線形計画問題の最適解に収束
- DMW-Game: コスト最小化ゲームの粗相関均衡点に収束

提案手法の利点

- 局所的な通信のみで動作
- DMW-Game: 効率的に良質な解が得られる
- DMW-LP: 下界が得られる

課題

- “堅い”制約，大域的な制約に対応