Online matching in large (random) graphs: theoretical and practical challenges

Matthieu Jonckheere

Stochastic Matching Workshop 2024

Session: Online Matching on Random Graphs

- Online matching on stochastic block models (SBM) (this talk)
- Online matching on geometric random graph (Flore)
- Matching on dynamic graphs (Aditi)

Outline and Co-authors

- First part on online matching models and theoretical challenges. Joint work with Pascal Moyal (Université de Lorraine), Claudia Ramirez and Nahuel Soprano-Loto (INRIA, Paris)

- Practical Reinforcement Learning approach to online matching. Joint work with Chiara Mignacco and Gilles Stoltz (INRIA, Orsay)

Two online matching models

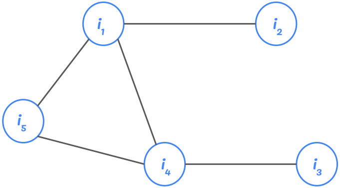
- Online/Dynamic matching on a fixed network model
- Online matching for a stochastic block model.

Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$

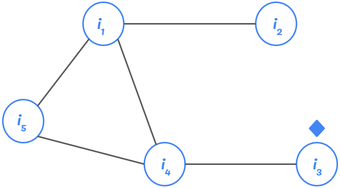
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



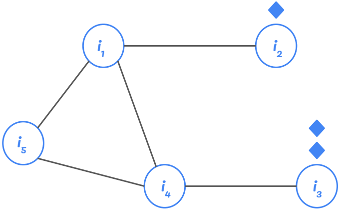
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



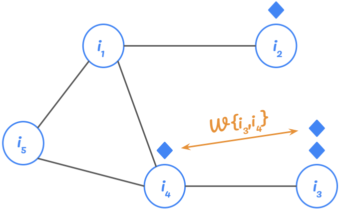
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



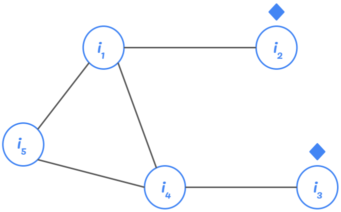
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



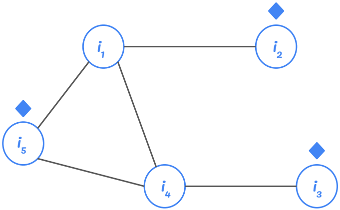
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



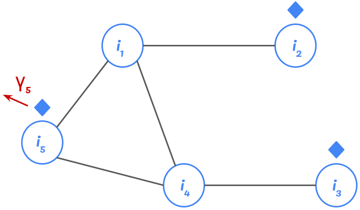
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



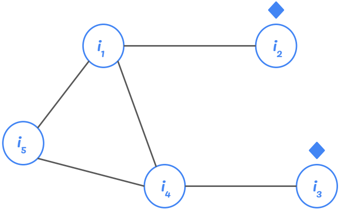
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



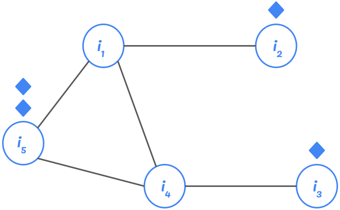
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



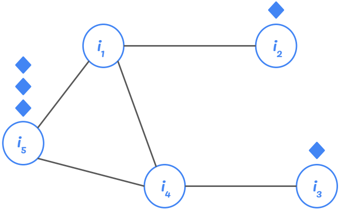
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



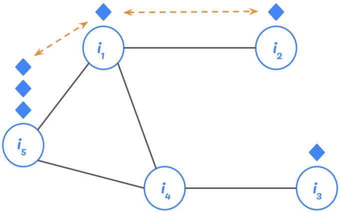
Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



Network model

- Compatibility's graph: $G = (V, E)$, where V are the individuals classes and E tells us that if $\{i, j\} \in E$ for two classes $\{i, j\} \in V$, then their are compatible.
- Arrival rates vector: $(\lambda_i)_{i \in V} \in (0, \infty)^V$
- Departure rates vector: $(\gamma_i)_{i \in V} \in [0, \infty)^V$
- Rewards vector: $(\omega_{\{i,j\}})_{\{i,j\} \in E} \in [0, \infty)^E$



Previous work

Bipartite stochastic matching model

- Caldentey et al. (2009)
- Adan and Weiss (2012)
- Bušić et al. (2013)

Compatibility's general graph

- Mairesse and Moyal (2016)

Generalized Max-weight policy

$V_{x,j}(i) =$ probability than given a state x and an arrival at node j, the policy decides to match an individual in node i.

$V_{x,j}$ is uniform in $\underset{i \in E(j): x(i) > 0}{argmax} \left(\eta[x(i) + \epsilon_{i,j}]^+ + \omega_{i,j}\right)$

NCOND

Let $R = \{i \in V : \gamma_i > 0\}$ the set of sites where there is impatience. The pair (λ, γ) will satisfy *NCOND* if

$$\lambda(I) < \lambda(E(I)),$$

for any I that is an independent subset of $V \setminus R$.
For $W \subset V$ we define $\lambda(W) = \sum_{i \in W} \lambda_i$, where $\lambda(W)$ is the total arrival rate to W.

Note. This condition was identified in *Mairesse and Moyal (2016)* as a necessary natural condition for the stability of a large family of policies in models without impatience.

More precise stability results

Theorem [Moyal, J., Soprano-Loto, Ramirez, 2022]
If the pair (λ, γ) satisfies *NCOND*, under max-weight, the function $f_2(x) = \sum_{i \in V} x(i)^2$ is a Lyapunov function.

Let π be the only stationary distribution.

Corollary
There are constants $\alpha, c > 0$ and $0 < \rho < 1$ such that $d_{TV}(\mathbb{P}_x(X_t \in \cdot), \pi) \leq c\rho^t e^{\alpha \|x\|_\infty}$.

Second model

Second-model

Multiclass matching on random graphs.

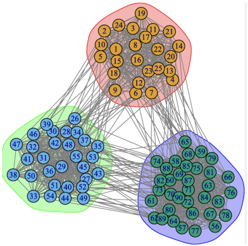
- Offline version: Given the graph find the maximal size matching (using compatibility rules)
- Online version: Given a random sequence of nodes, match them given only the information of already matched nodes.

Perfect Matching

Single class: a subset of the edges such that every vertex in the graph is incident to exactly one edge from this subset.

Multi-class: a subset of the edges such that every vertex in the graph is incident to exactly one **compatible** edge from this subset.

A stochastic block model (SBM) with p communities



A random graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$:

- There are p communities $\mathbf{C}^1, ..., \mathbf{C}^p$ forming a partition of $\mathbf{V}$.
- For any nodes $u_i \in \mathbf{C}^1$ and $u_j \in \mathbf{C}^j$, the edge $\{u_i, u_j\} \in \mathbf{E}$ with probability P_{ij}, independently of everything else.
- Set G, the *root* graph (with self-loop) on the set of nodes $[1, p]$, such that for all $i, j \in [1, p]$,

$$i \sim j \iff P_{ij} > 0.$$

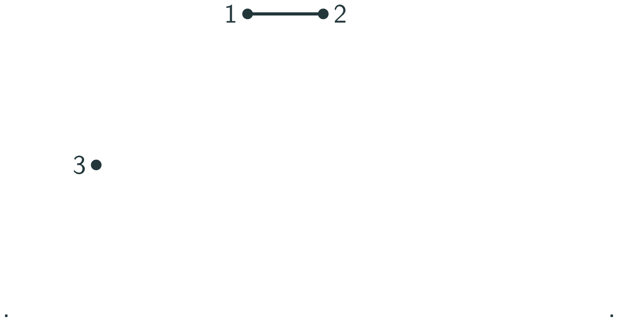
Joint construction of an online matching on the SBM

p=4, nodes of the graphs "arrive sequentially",

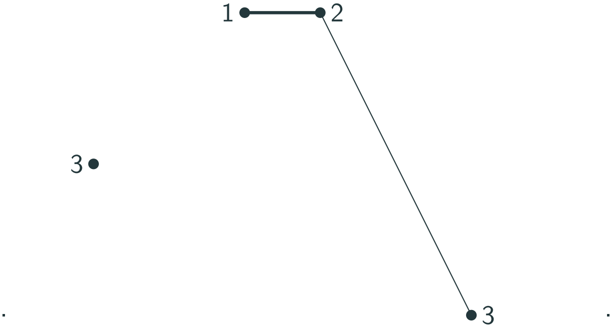
We indicate the class of the node.

Joint construction of an online matching on the SBM

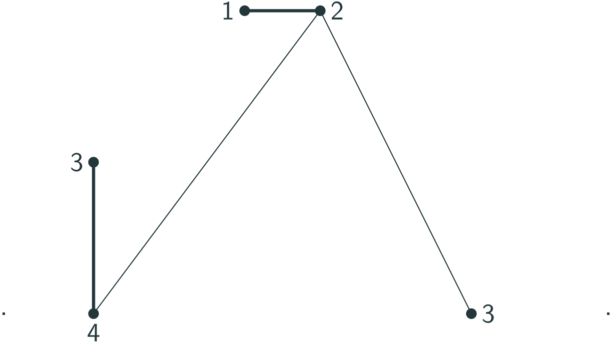
Joint construction of an online matching on the SBM



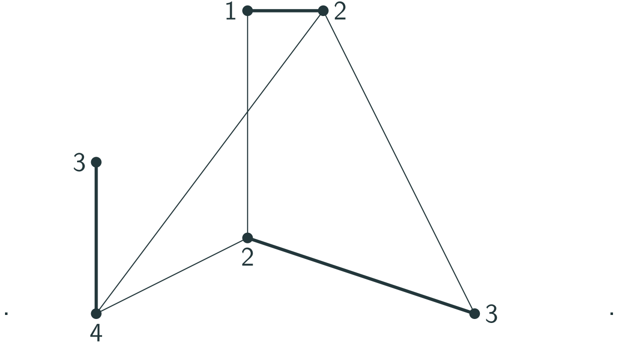
Joint construction of an online matching on the SBM



Joint construction of an online matching on the SBM



Joint construction of an online matching on the SBM



Markovian description

- Given $(\lambda_i)_{i=1\dots p}$ the arrival probabilities, (proportion of nodes of each class),
- we define the process

$$Q_n := (|\mathbf{Q}_n(1)|, \cdots, |\mathbf{Q}_n(p)|),$$

as the number of unmatched items of each class at time n. Then, Q_n is an irreducible Markov DTMC for "Markovian policies".

Matching policy

Consider the following matching criterion. At step n,

1. If the incoming item v_n is of community $\mathbf{C}^j$, set the next match as

$$\text{uniform in } \operatorname{Argmax}\{x(i)\,:\, x(i) > 0 \text{ and } i \sim j\}.$$

2. Then,
    - If v_n indeed shares an edge with some node $u_n \in \mathbf{C}_j$ add the edge $\{u_n, v_n\}$ to $\mathbf{M}_{n-1}$.
    - Else, leave $\mathbf{M}_n = \mathbf{M}_{n-1}$.

Matching policy

Observe that:

- The matching $\mathbf{M}_n$ is perfect if and only if

$$(|\mathbf{Q}_n(1)|, \cdots, |\mathbf{Q}_n(p)|) = \mathbf{0}.$$

$\Rightarrow$ The matching is perfect infinitely often if and only if the Q_n is positive recurrent.

Results

Define $Q(n) = (Q_1(n), \cdots, Q_L(n))$ the Markov chain describing the number of unmatched items at step n of each class.

Theorem (Soprano-Loto,J, Moyal)
Q is positive recurrent **if and only if** $\lambda \in \textsc{Ncond}(G)$;

1. If $\lambda \in \textsc{Ncond}(G)$ then the matching is a.s. perfect infinitely often;
2. Bounds on the expected number of unmatched nodes, to the large-graph limits.

Idea of the proof

- We can show that both Markovian dynamics (Model I and II) are very close.
- Using our previous insights on Model I, we can prove that a sum of square is again Lyapunov.

Connected and open problems

- Optimality gaps for other regimes (mean number of connections per node of order n^{γ}, $\gamma < 1$).

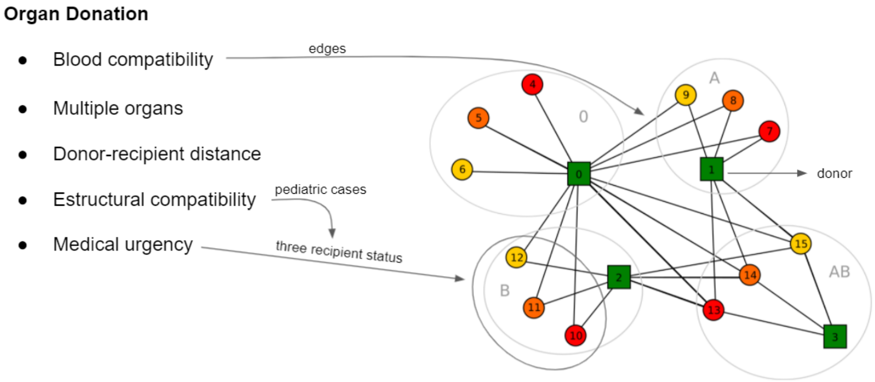
- For $\gamma = 0$, (sparse case), the optimality gap characterized for 1 class.

Intermezo towards applications

Application: Study of UNOS data

- The United Network for Organ Sharing (UNOS) has a detailed database recording the last 31 years of organ donation in the United States. These data is available on the OPTN platform
- We used a subset of data, the heart case - we assume that these cases occur without taking into consideration distance, given the short time living outside the body - differentiating blood types (AB0 system) and urgency status (H1A, H1B and H2) for pediatric cases - in order to work with fewer urgency status (adults have 10 different status),
- all of these filters were applied in the data, except for some specific parameters where we used a value for adults and pediatric cases together.

Application: System



Organ donation graph for pediatric cases differentiated by urgency status for recipients -status H1A, H2A and H2 in red, orange and yellow respectively- and blood type -circled in gray- for recipients (circles) and donors (squares).

Application: System

In order to adjust our model of Markov processes to these data we need to fit 5 values:

1. Arrival rate for donors, λ_D.
2. Arrival rate for recipients, λ_R.
3. Departure rate for donors, γ_D.
4. Departure rate for recipients, γ_R.
5. Number of elements per node in the initial system, WL_i.

Application: Parameters

- Actual waiting list by edge, blood type and urgency status.
- Transplants by year, blood type and urgency status.
- Additions to WL by year, blood type and urgency status.
- Deaths by year, blood type and urgency status.
- Median waiting time for a transplant by blood type or age or urgency status.
- ...

	0	A	B	AB
Heart Status 1A	43	25	13	4
Heart Status 1B	33	24	9	3
Heart Status 2	70	26	17	1

	0	A	B	AB
2016	202	173	60	9.0
2017	213	142	64	12.0
2018	221	155	69	23.0
2019	187	165	61	20.0

median/365d

	0	A	B	AB
λ_D	0.59*s	0.41*s	0.18*s	0.05*s

	0	A	B	AB
Heart Status 1A	173,5	125,5	51,5	12
Heart Status 1B	65,5	44,5	20	3,5
Heart Status 2	56,5	38,5	14,5	4,5

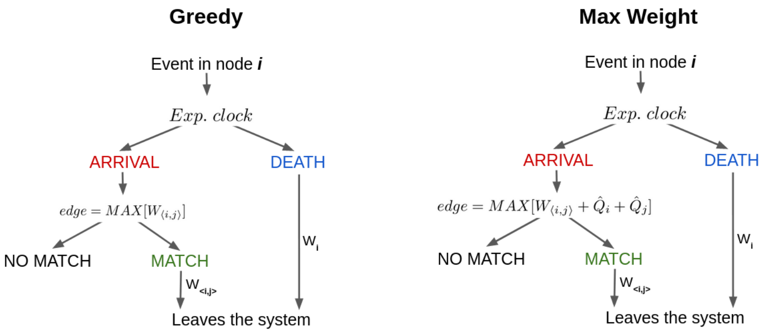
1/365d

	0	A	B	AB
Heart Status 1A	0.51	0.37	0.14	0.04
Heart Status 1B	0.20	0.12	0.05	0.01
Heart Status 2	0.15	0.14	0.04	0.01

	WTT	γ_R
Heart Status 1A	f*87	0.01/f
Heart Status 1B	f*253	0.004/f
Heart Status 2	f*726	0.001/f

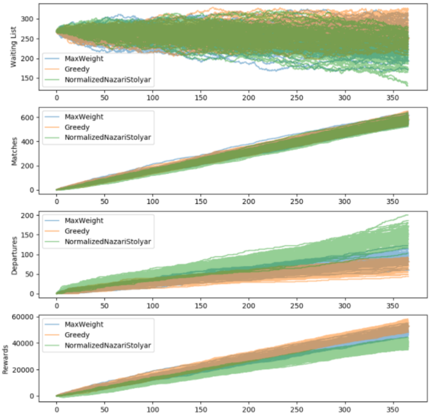
https://optn.transplant.hrsa.gov/data

Application: Policies

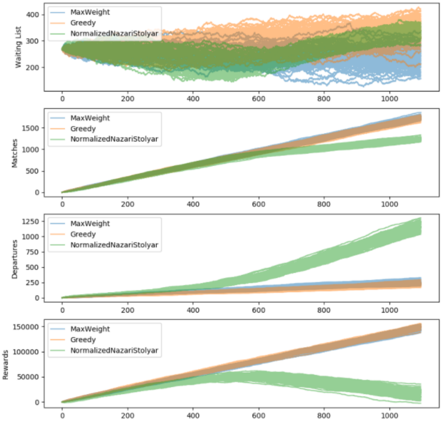


Application: Results

~1,200 arrivals
~1 year

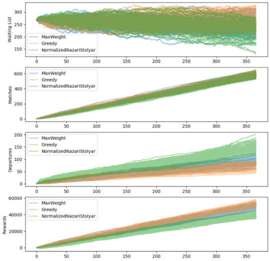


~3,500 arrivals
~3 years

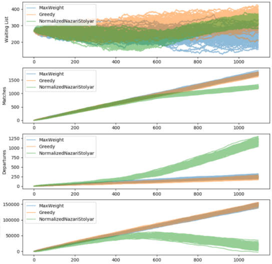


Application: Results

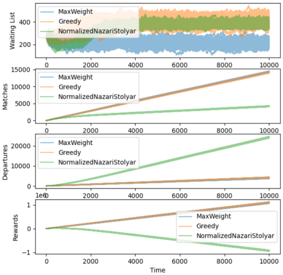
~1,200 arrivals
~1 year



~3,500 arrivals
~3 years



~35,000 arrivals
~30 years



What are we looking for?

- **Interpretable policies**
- Efficient policies at an intermediate time-scale.
- **Robust policies**

Proposal: Reinforcement learning orchestration between experts policies.

Reinforcement Learning approach

- Markov Decision Processes (MDPs):
  - Finite state space S and action space A.
  - Transition kernel $T : S \times A \to P(S)$.
  - Reward function $R : S \times A \to P([0, 1])$.
- Objective: Learn a policy π that maximizes the expected sum of discounted rewards.

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t) \mid s_0 = s\right]$$

- Optimal policy π^{*} maximizes the value function $V^{\pi}(s)$.

Challenges in RL

- Large state or action spaces lead to prohibitively slow learning.
- Developing efficient algorithms to handle these spaces is crucial for practical applications.
- Need for faster algorithms to ensure better performance.
- Existing methods struggle with model-free RL and large MDPs.

Orchestration of Expert Policies

Orchestration of RL Policies

- A collection $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$ of expert policies.
- Combine these policies using state-dependent weights $q_k(s)$.

$$q_\Pi(a|s) = \sum_{k=1}^{K} q_k(s)\pi_k(a \mid s)$$

- Learn a policy q_Π in this class as close as possible to q_Π^*.

Advantage Functions

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\right]$$

- Use of Advantage Functions $A^\pi(s,a)$ to improve policy construction:

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

Orchestration Strategy: Weight Calculation

- Sequential strategy ϕ:

$$q_t(\cdot|s) = \phi_t(\sum_{l \leq t} \hat{A}_l^\pi(s, \cdot))$$

- Examples of ϕ: $\phi_t(x) = e^{\eta_t x}$, $\phi(x) = e^{\eta x}$, $\phi(x) = \max(x, 0)^p$...
- Exponential potential-based methods are often used to update weights (Cai et al. 2020, Shani et al. 2020).
- When ϕ is exponential, we obtain (almost) the natural policy gradient.

Regret Definition and Bound (without estimation)

- Regret:

$$V^*(s) - V_{q_t}(s) = (V^*(s) - V^*_{\Pi}(s)) + \left(V^*_{q_t}(s) - V^*_{\Pi}(s)\right)$$

- We transfer bounds from adversarial learning to RL.

Theorem (J., Mignacco, Stoltz).

$$\forall s \in S, \forall T \geq 0 : V^*_\Pi(s) - \frac{1}{T}\sum_{t=1}^{T} V_{q_t}(s) \leq \frac{\sqrt{\log K}}{(1-\gamma)^2\sqrt{T}}$$

Experiments

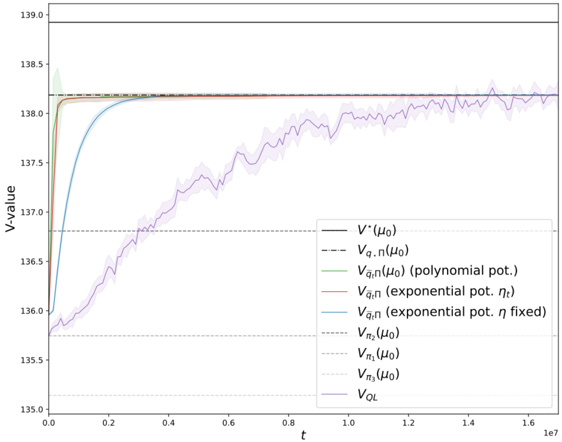


Figure 1: Small network

References

1. Bušić A., Gupta V., Mairesse J. (2013). *Stability of the bipartite matching model.* Adv. in Appl. Probab. 45(2):351-378.
2. Mairesse J. Moyal P. (2016). *Stability of the stochastic matching model.* J. Appl. Probab. 53(4):1064-1077
3. Jonckheere M., Moyal P., Ramirez C., Soprano-Loto N. (2022). *Generalized max-weight policies in stochastic matching.* Stochastic Systems 13 (1), 40-58
4. Soprano-Loto N., Jonckheere M., Moyal P. (2023) *Online matching for the multiclass stochastic block model*, arXiv preprint arXiv:2303.15374
5. Jonckheere, M, Mignacco C., Stoltz G., (2024). *Symphony of experts: orchestration with adversarial insights in reinforcement learning* arXiv preprint arXiv:2310.16473

Thank you !