An Online Learning Approach to a Multi-player N-armed Functional Bandit

Sam O'Neill[0000−0003−2926−6921], Ovidiu Bagdasar[0000−0003−4193−9842], and Antonio Liotta[0000−0002−2773−4421]

University of Derby, Kedleston Rd, Derby, DE22 1GB, United Kingdom
s.oneill@derby.ac.uk, o.bagdasar@derby.ac.uk, a.liotta@derby.ac.uk

Abstract. Congestion games possess the property of emitting at least one pure Nash equilibrium and have a rich history of practical use in transport modelling. In this paper we approach the problem of modelling equilibrium within congestion games using a decentralised multi-player probabilistic approach via stochastic bandit feedback. Restricting the strategies available to players under the assumption of bounded rationality, we explore an online multiplayer exponential weights algorithm for unweighted atomic routing games and compare this with a ϵ-greedy algorithm.

Keywords: Congestion games · Online learning · Multi-armed bandit

1 Introduction

The multi-armed bandit (MAB) problem has received much attention in recent years within the online and machine learning community due to its appropriateness for demonstrating the fundamental trade-off between exploration and exploitation in online learning. The basic MAB problem is for an agent to maximise the cumulative reward received after playing a number of rounds (finite or infinite). In each round the agent is required to choose one of K bandits and receives an associated reward. For an agent to be successful it must employ a strategy which balances the trade-off between exploration and exploitation. Explore too little and the agent's preferred choice may remain sub-optimal, explore too often and the agent fails to exploit the most optimal choices. Numerous algorithms have been studied for variants of the MAB problem and a popular measure of an algorithm's performance is the notion of expected regret, whereby the agent's received reward is compared with the expected reward that would have been received for the optimal choices [1].

In strategic repeated games, a natural approach towards equilibrium is to employ an online learning algorithm in which the expected regret of the player(s) is minimised over the time horizon [2]. Whilst expected regret analysis and convergence of equilibrium are important and rich areas of research, they make some key assumptions that could, in certain modelling scenarios, be deemed too restrictive. First, when bounding the regret of an algorithm it is necessary that the

utility received by a player is itself bounded, therefore restricting the types of utility function. Second, convergence to a state of equilibrium does not take into account the capricious nature of certain individuals and that a player's rationality is often bounded by both the intractability of the decision making process and the player's preference for exhaustive search [4]. Therefore the best one may be able to do is express a player's belief in the most preferable choices over a set of tractable strategies.

The above concepts are particularly inherent in routing games, a form of strategic repeated game in which multiple players (e.g. drivers of vehicles) simultaneously route flow across a network in an attempt to minimise their own cost. Routing games belong to the larger class of congestion games which possess the property of emitting at least one pure strategy Nash equilibrium [6] and have received much attention within the field of algorithmic game theory [7]. However, due to the underlying graph structure, the strategy set for these games suffers from the "curse of dimensionality" whereby the strategy set for a source sink pair (available paths) grows exponentially with the size of the underlying graph. Traditionally methods have employed a centralised approach in which full information of the costs associated with all strategies is known, and flow is shifted globally between paths so as to satisfy a set of constraints representing a state of equilibrium for the given problem [5]. Such approaches fail to consider both the decentralised nature of the decision making processes within the system and that individual players have a particularly myopic view of the system and, therefore, tend to make decisions on very little information.

Motivated by the concepts of bounded rationality and random/deliberate sub-optimal choices, the focus of this paper is to model unweighted atomic routing games under a restricted subset of strategies via noisy feedback, i.e. the utility may vary due to external factors. We investigate an exponential weights algorithm which at each time step (round) uses feedback as a mechanism for a player to update their personal beliefs (probability distribution) of the best course of action and an ϵ-greedy algorithm in which the best course of action is selected greedily with probability $p = 1 - \epsilon$. Variants of both algorithms are implemented for the semi-bandit and bandit feedback scenarios.

2 Preliminaries

2.1 Congestion Games

An N-player congestion game consists of a finite number of players $\mathcal{N} = \{1, \cdots, N\}$, a set of congestible elements $\mathcal{E}$ with associated cost (latency) functions $l_e : \mathbb{N} \mapsto \mathbb{R}$ for each element $e \in \mathcal{E}$ and a set of playable strategies $\mathcal{A}_i$ for each player i, where a given strategy $a_i \in \mathcal{A}_i$ is a set of congestible elements $a_i \subseteq \mathcal{E}$. The number of players choosing element e is $x_e = \sum_{i \in \mathcal{N}} \sum_{e_i \in a_i} \mathbb{1}(e_i = e)$, where $\mathbb{1}$ is the indicator function. The associated cost to player i playing strategy a_i is $u_i(a_i; a_{-i}) = \sum_{e \in \mathcal{E}} \sum_{e_i \in a_i} \mathbb{1}(e_i = e) \cdot l_e(x_e)$.[1] That is each player picks a set of

[1] $(a_i; a_{-i})$ is commonly used to refer to player i's strategy given the strategy profile $\mathbf{a} = (a_1, \cdots, a_i, \cdots, a_N)$.

congestible elements and their associated costs are dependent not only on their own strategy, but on those played by the other players. The total cost U under strategy profile $\mathbf{a} = (a_i)_{i \in \mathcal{N}}$ is then,

$$U(\mathbf{a}) = \sum_{i \in \mathcal{N}} u_i(a_i; a_{-i}) = \sum_{i \in \mathcal{N}} \sum_{e \in \mathcal{E}} \sum_{e_i \in a_i} \mathbb{1}(e_i = e) \cdot l_e(x_e) = \sum_{e \in \mathcal{E}} x_e l_e(x_e).$$

Let $\mathcal{A} = \prod_i \mathcal{A}_i$ to be the set of all strategy profiles and $l = (l_e)_{e \in \mathcal{E}}$ the vector of cost functions associated with each e, then the congestion game is described by the tuple $(\mathcal{N}, \mathcal{E}, \mathcal{A}, l)$.

Rosenthal showed that a congestion game has at least one pure strategy Nash equilibrium found by minimising the potential function $\Phi = \sum_{e \in \mathcal{E}} \sum_{i=1}^{x_e} l_e(x_e)$ [6].

2.2 Unweighted Atomic Routing Game

For an unweighted atomic routing game, let the set of congestible elements $\mathcal{E}$ be the edges in the graph $G = (V, E)$ and for each player $i \in \mathcal{N}$ associate a source/sink pair (o_i, d_i) and traffic demand $k_i = 1$, i.e. players route themselves.[2] A player's strategy set $\mathcal{A}_i$ is the set of possible paths from source to sink, i.e. a strategy $a_i \in \mathcal{A}_i$ is a path consisting of edges $e \in E$ [7]. Therefore the cost to a given player choosing a particular path is dependent on the number of players choosing paths which share edges in the graph.

As a bandit problem, an unweighted atomic routing game consists of $\mathcal{N}$ players, a set E of functional bandit machines (edges), with corresponding congestion functions l. Each player $i \in \mathcal{N}$ then pulls a combination of bandit machines $a_i \subseteq E$ (path) from the strategy set $\mathcal{A}_i$ (set of available paths for (o_i, d_i) pair) and receives feedback given the strategy profile of played actions $\mathbf{a} = (a_i)_{i \in \mathcal{N}}$.

3 Learning Under Bandit Feedback

The following section introduces the exponential weights and ϵ-greedy algorithms for both semi-bandit and bandit feedback.

For each player i let $W_i^t = (W_{ia_i}^t)_{a_i \in \mathcal{A}_i}$ be a set of weights associated with the player's available strategies at a given round t. We denote the probability of a player selecting strategy a_i as,

$$\mathbf{P}_{ia_i}^t = \frac{W_{ia_i}^t}{\sum_{j=1}^{|\mathcal{A}_i|} W_{ij}^t},$$

and the probability distribution over all strategies $\mathcal{A}_i$ as $\mathbf{P}_i^t = (\mathbf{P}_{ia_i}^t)_{a_i \in \mathcal{A}_i}$.

[2] In general an unweighted traffic rate routes the same quantity $d_i = d \quad \forall i \in \mathcal{N}$

3.1 Semi-bandit Feedback

Under semi-bandit feedback, the player has access to the entire payoff vector of playable strategies. The noisy feedback for a given strategy a_i^t played by player i in round t is,

$$\hat{r}_{ia_i}(a_{-i}^t) = u_i(a_i^t; a_{-i}^t) + \xi_{ia_i}^t,$$

and the entire payoff vector for all strategies available to player i is then

$$\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t} = (\hat{r}_{ia_i}(a_{-i}^t))_{a_i \in \mathcal{A}_i}$$

For each player i, the exponential weights algorithm (see Algorithm 1) maintains the probability distribution $\mathbf{P}_i^t = (\mathbf{P}_{ia_i}^t)_{a_i \in \mathcal{A}_i}$ reflecting the beliefs about player i's best strategy from the strategy set $\mathcal{A}_i$. At time t, player i samples an action $a_i^t \sim \mathbf{P}_i^t$ and updates the distribution $\mathbf{P}_i^{t+1}$ based on the semi-bandit feedback it receives [3]. Note that due to the interdependence of the congestion functions, all players actions must be selected and played before players receive their corresponding feedback.

Algorithm 1 Exponential weights with semi-bandit feedback [EW-SB]

Require: $\gamma_t = t^{-\frac{1}{\alpha}}\ \forall t \in [1, \ldots, T]$, $W_i^0 \in \mathbf{1}^{|\mathcal{A}_i|}\ \forall i \in \mathcal{N}$

for $t = 1, \ldots, T$ **do**
  for each player i in $\mathcal{N}$ **do**
    $\mathbf{P}_i^t = \frac{W_i^t}{\sum_{j=1}^{|\mathcal{A}_i|} W_{ij}^t}$ ▷ Calculate probability distribution for strategies
    $a_i^t \sim \mathbf{P}_i^t$ ▷ Sample action from probability distribution
  end for
  for each player i in $\mathcal{N}$ **do**
    $\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t} = (\hat{r}_{ia_i}(a_{-i}^t))_{a_i \in \mathcal{A}_i}$ ▷ Observe estimated reward for strategies
    $W_i^{t+1} = W_i^t \cdot \exp\left(\frac{\gamma_t \hat{\mathbf{r}}_\mathbf{i}^\mathbf{t}}{|\mathcal{A}_i|}\right)$ ▷ Update weights
  end for
end for

The ϵ-greedy algorithm (see Algorithm 2) updates the average reward for all player strategies via the feedback vector and greedily selects the best known strategy with probability $p = 1 - \epsilon$ and randomly selects an action with probability $p = \frac{\epsilon}{\mathcal{A}_i}$.

3.2 Bandit feedback

Under bandit feedback the player only has access to feedback for the strategy played in round t and therefore a player must attempt to estimate the cost of strategies over time. The exponential Weights algorithm can be amended (see Algorithm 3) by utilising the importance sampling estimator.

Algorithm 2 ϵ-greedy with semi-bandit feedback [ϵG-SB]

Require: $W_i^0 \in \mathbf{0}^{|\mathcal{A}_i|}\ \forall i \in \mathcal{N}$
for $t = 1, \ldots, T$ **do**
  for each player i in $\mathcal{N}$ **do**
    if $\epsilon_t \sim unif(0,1) < \epsilon$ **then**
      $a_i^t \sim unif\{1, |\mathcal{A}_i|\}$ ▷ Choose at random with probability $p = \frac{1}{|\mathcal{A}_i|}$
    else
      $a_i^t = \arg\max_{a_i \in \mathcal{A}_i}(W_{ia_i}^t)$
    end if
  end for
  for each player i in $\mathcal{N}$ **do**
    $\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t} = (\hat{r}_{ia_i}(a_{-i}^t))_{a_i \in \mathcal{A}_i}$ ▷ Observe estimated feedback for strategies
    $W_i^{t+1} = W_i^t + \frac{1}{t+1}\left[\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t} - W_i^t\right]$ ▷ Update average feedback
  end for
end for

The feedback for strategy a_i^t received in round t is the individual cost incurred by the player,

$$\hat{u}_i^t = u_i(a_i^t; a_{-i}^t) + \xi_i^t$$

and the full feedback vector $\mathbf{r_i}(a_{-i}^t)$ can be estimated by,

$$\hat{r}_{ia_i}^t = \begin{cases} \frac{\hat{u}_i^t}{\mathbf{P}_{ia_i}^t}, & \text{if } a_i = a_i^t. \\ 0, & \text{otherwise.} \end{cases} \quad \forall a_i \in \mathcal{A}_i.$$

It can be shown [3] that under certain probabilistic assumptions , $\hat{r}_{ia_i}^t$ results in an unbiased estimator of the feedback received by player i playing action a_i calculated over the joint probability of all other strategy profiles $a_{-i} \in \prod_{j \neq i} \mathcal{A}_j$,

$$\mathbf{P}_{-i}^t = (\mathbf{P}_{-ia_{-i}}^t)_{a_{-i} \in \mathcal{A}_{-i}},$$

namely,

$$\mathbb{E}_t[\hat{r}_{ia_i}^t] = u_i(a_i; \mathbf{P}_{-i}^t) = \sum_{a_{-i} \in \mathcal{A}_{-i}} \mathbf{P}_{-ia_{-i}}^t u_i(a_i^t; a_{-i}^t).$$

For the ϵ-greedy algorithm (see Algorithm 4) we amend the update of the average rewards W_i^{t+1} to only update the strategy that has been played at time t.

4 Preliminary Results

Algorithms $1-4$ were tested on a bidirectional lattice network with 16 vertices and 48 edges. Given the stochastic nature of the algorithms, 10 randomly generated instances of the lattice network were generated and 250 players were routed between 4 origin destination pairs. The results were averaged over 10 episodes

Algorithm 3 Exponential weights with bandit feedback [EW-B]

Replace lines $7-8$ in algorithm 1 with:

$\hat{u}_i^t = u_i(a_i^t; a_{-i}^t) + \xi_i^t$ ▷ Observe estimated feedback for played strategy

$$\hat{r}_{ia_i}^t = \begin{cases} \frac{\hat{u}_i^t}{\mathbf{P}_{ia_i}^t}, & \text{if } a_i = a_i^t. \\ 0, & \text{otherwise.} \end{cases} \quad \forall a_i \in \mathcal{A}_i$$

▷ Estimate feedback vector $\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t}$

$W_i^{t+1} = W_i^t \cdot \exp\left(\frac{\gamma_t \hat{\mathbf{r}}_\mathbf{i}^\mathbf{t}}{|\mathcal{A}_i|}\right)$ ▷ Update weights

Algorithm 4 ϵ-greedy with bandit feedback [ϵG-B]

Replace lines $10-11$ in algorithm 2 with:

$\hat{u}_i^t = u_i(a_i^t; a_{-i}^t) + \xi_i^t$ ▷ Observe estimated feedback for played strategy

$$\hat{r}_{ia_i}^t = \begin{cases} \frac{1}{t+1}\left[\hat{u}_i^t - W_{ia_i}^t\right], & \text{if } a_i = a_i^t. \\ 0, & \text{otherwise.} \end{cases} \quad \forall a_i \in \mathcal{A}_i$$

▷ Estimate feedback vector $\hat{\mathbf{r}}_\mathbf{i}^\mathbf{t}$

$W_i^{t+1} = W_i^t + \hat{\mathbf{r}}_\mathbf{i}^\mathbf{t}$ ▷ Update average rewards

per network - each episode consisting of a 100 rounds ($T = 100$).[3] Figure 1 (a) plots the total cost U averaged over the data set and for comparison, the total cost U_Φ experienced at the equilibrium given by the potential function Φ. Figure 1 (b) plots the regret of each algorithm defined to be the cumulative sum of the difference between the total cost of the played strategy profile $\mathbf{a}^t$ at time t and the equilibrium total cost U_Φ,

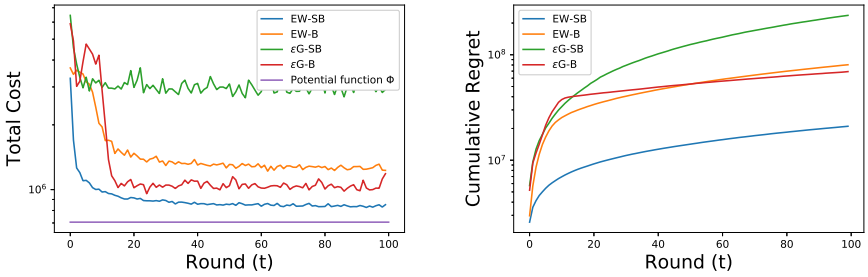
$$R_t = \sum_{i=t}^{t} \left[U(\mathbf{a}^\mathbf{t}) - U_\Phi\right].$$

Finally figure 2 plots the individual costs for the players at the initial and the final (T) round. Clearly a more uniform cost has emerged at time T for the 4 origin/destination pairs and this compares well with the costs at equilibrium given by minimising Φ (indicated in red).

5 Concluding Remarks

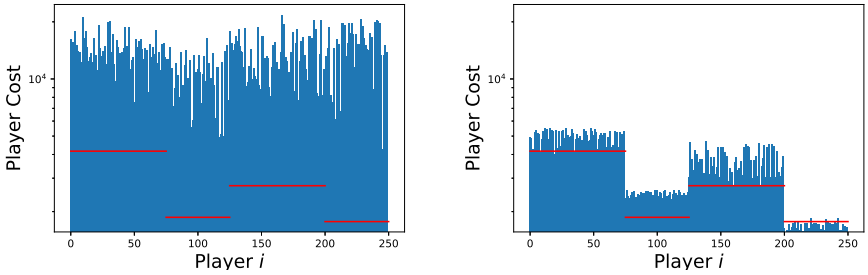
On average, the exponential weights algorithm with semi-bandit feedback performs the best over the data set and compares reasonably well with the total

[3] The source code is available at https://github.com/samtoneill/congestionbanditgames



(a) Total Cost (b) Cumulative Regret

Fig. 1: Log-lin plots of total cost and cumulative regret for the 4 algorithms averaged over all test data



(a) Player costs after initial round (b) Player costs after T rounds

Fig. 2: Log-lin plots illustrating the convergence of players costs for the 4 origin/destination pairs

cost associated with the Nash equilibrium given by minimising Φ. It is worth noting that the two bandit feedback algorithms which, arguably, in certain circumstances represent a more realistic model, e.g. a player would only experience or log their own travel time, perform comparably well. The poor performance of the ϵ-greedy algorithm with semi-bandit feedback is also of interest and, while it would require more investigation, a possible cause is that certain strategies $a_i \in \mathcal{A}_i$ experience an extreme cost under certain strategy profiles $(a_i; a_{-i})$ and therefore the averages maintained become unrepresentative of the more optimal choices.

Whilst it can be argued that these algorithms more realistically represent a player's decision making processes when taking into account human nature, they are not designed to be efficient in terms of computational complexity and therefore they may not be practical for use on larger networks. A future direction would be to employ similar techniques using a more scalable function approximation, such as a neural network, to keep track of a player's beliefs. There is also the possibility of using reinforcement learning techniques to employ autonomous

agents whose primary role is to act altruistically for the benefit of the other agents within the network to reduce the overall congestion experienced [8].

References

1. Belmega, E.V., Mertikopoulos, P., Negrel, R., Sanguinetti, L.: Online convex optimization and no-regret learning: Algorithms, guarantees and applications (4 2018), http://arxiv.org/abs/1804.04529
2. Cesa-Bianchi, N., Lugosi, G.: Prediction, learning, and games. Cambridge University Press (2006)
3. Cohen, J., Héliou, A., Mertikopoulos, P.: Learning with bandit feedback in potential games (12 2017), https://hal.archives-ouvertes.fr/hal-01643352
4. Gigerenzer, G., Selten, R.: Bounded rationality : the adaptive toolbox. MIT Press (2001)
5. Patriksson, M.: The traffic assignment problem : models and methods. Dover Publications (1994)
6. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory **2**(1), 65–67 (12 1973). https://doi.org/10.1007/BF01737559
7. Roughgarden, T.: Routing Games. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (eds.) Algorithmic Game Theory, pp. 461–486. Cambridge University Press, Cambridge (2007). https://doi.org/10.1017/CBO9780511800481.020
8. Vinitsky, E., Kreidieh, A., Flem, L.L., Kheterpal, N., Jang, K., Wu, C., Wu, F., Liaw, R., Liang, E., Bayen, A.M.: Benchmarks for reinforcement learning in mixed-autonomy traffic. In: Billard, A., Dragan, A., Peters, J., Morimoto, J. (eds.) Proceedings of The 2nd Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 87, pp. 399–409. PMLR (5 2018), http://proceedings.mlr.press/v87/vinitsky18a.html