Distributed Load Balancing in Heterogeneous Networks

Mohd. Shabbir Ali, Pierre Coucheney, and Marceau Coupechoux
TELECOM ParisTech (INFRES), UVSQ/PRiSM

ANR NETLEARN http://netlearn.enst.fr/
Info : Workshop on Learning and Networks 9th Oct 2015 in Orange Labs

10 Apr. 2015

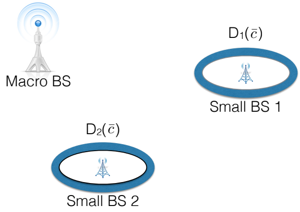
Outline

- Motivation
- Heterogeneous network model
- Cell range expansion (CRE) association rule
- CRE game formulation
- Distributed learning algorithms
- Simulation results
- Conclusion and future work

Motivation

- Heterogeneous networks: cellular networks are densified with small cells
- Due to the transmit power differences, there can be a high imbalance between stations
- Cell Range Expansion is a technique to increase the serving area of small cells
- We look for a scheme as distributed as possible and rely on potential game framework
- Our aim is not to study the interactions between competiting selfish users but to enforce through utility design the potential game structure

System Model I



- $\mathcal{S}$ set of base stations (BSs)
- P_i transmit power of BS i
- $g_i(x)$ channel gain of BS i at location x
- $\bar{c} = \left[c_1, c_2, \ldots, c_{|\mathcal{S}|}\right]$ CRE vector
- $\rho = \left[\rho_1, \rho_2, \ldots, \ \rho_{|\mathcal{S}|}\right]$ load vector

CRE Association Rule

- Associate user to the BS that provides highest biased received power

$$\mathcal{D}_i(\bar{c}) = \{x | \forall j \in \mathcal{S}, P_i g_i(x) c_i \geq P_j g_j(x) c_j, \gamma_i(x) \geq \gamma_{\min}\}. \quad (1)$$

- Load of BS i

$$\rho_i(\bar{c}) = \int_{x \in \mathcal{D}_i(\bar{c})} \varrho_i(x) dx, \quad (2)$$

$$\varrho_i(x) = \frac{\lambda(x)}{\mu(x) \upsilon_i(x)}. \quad (3)$$

- $\lambda(x)$ is flow arrival rate per unit area [arrivals/s/m^2]
- $\frac{1}{\mu}$ is average file size [bits]
- $\upsilon_i(x)$ is data rate [bits/s]

Objective function

Non-convex $\alpha-$fairness objective function [Kim et al., 2012]

$$\phi_\alpha(\bar{c}) = \begin{cases} \sum_{i\in\mathcal{S}} \frac{(1-\rho_i(\bar{c}))^{1-\alpha}}{\alpha-1}, & \alpha \geq 0, \\ -\sum_{i\in\mathcal{S}} \log\left(1-\rho_i(\bar{c})\right), & \alpha = 1, \end{cases} \tag{4}$$

$$\text{Feasible set } \mathcal{F} = \{\rho | 0 \leq \rho_i(\bar{c}) < 1, c_i \in [1, c_{\max}], \forall i \in \mathcal{S}\}. \tag{5}$$

Goal

Minimise $\phi_\alpha(\bar{c})$.

- $[\alpha = 0]$: Rate-optimal policy
- $[\alpha = 1]$: Proportional-fair policy
- $[\alpha = 2]$: Delay-optimal policy
- $[\alpha \to \infty]$: Min-max load policy

Example of Non-convex Objective Function

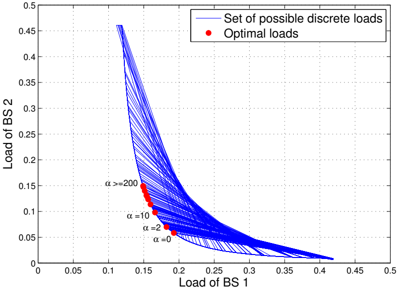


Figure : Feasible set $\mathcal{F}$ for 2 BSs.

- Even if the CRE set were continuous, $\mathcal{F}$ would not be convex.
- All the optimal load points are located on the Pareto frontier.
- The point for $\alpha \geq 200$ is the min-max load point.

CRE Selection Game

Definition (CRE selection game)

$\Gamma = \left(\mathcal{S}, \{X_i\}_{i\in\mathcal{S}}, \{U_i\}_{i\in\mathcal{S}}\right)$, *$\mathcal{S}$ is a set of BSs, $X_i = \{1, 1.1, 1.2, \ldots, 16\}$ is strategy set, and $U_i : X_1 \times X_2 \times \ldots \times X_{\mathcal{S}} \to \mathcal{R}$ is a cost function.*

Definition (Exact potential game [Monderer and Shapley, 1996])

If there exists $\phi_\alpha : X \to \mathcal{R}$ such that $\forall i \in \mathcal{S}$, $\forall c_i, c_i' \in X_i$ and $\forall c_{-i} \in X_{-i}$,

$$U_i(c_i, c_{-i}) - U_i(c_i', c_{-i}) = \phi_\alpha(c_i, c_{-i}) - \phi_\alpha(c_i', c_{-i}). \quad (6)$$

Theorem ([Monderer and Shapley, 1996])

Every finite potential game posses a pure-strategy Nash equilibrium (NE).

Theorem ([Monderer and Shapley, 1996])

All NE are contained in the set of minima of the potential function.

Cost Structure for Potential Game

- *Identical interest utility (IIU):* makes the network centralised.

$$U_i(\bar{c}) = \phi_\alpha(\bar{c}). \tag{7}$$

- *Wonderful life utility (WLU):* makes the network distributed.

$$U_i(c_i, c_{-i}) = \sum_{j \in N_i} \frac{(1 - \rho_j(c_i, c_{-i}))^{1-\alpha}}{\alpha - 1}. \tag{8}$$

where N_i is the neighbour set of BS i,

$$N_i = \bigcup_{\bar{c}} \{j \in \mathcal{S} | \exists x \in \mathcal{D}_i(\bar{c}), P_i g_i(x) c_i = P_j g_j(x) c_j\}. \tag{9}$$

A neighbour set N_i is all possible BSs that share boundary with BS i for at least one bias value.

Distributed Learning Algorithms

- *Complete information:* BS knows the cost of playing any action.
    - Best response (BR) algorithm.
    - Log-linear learning algorithm (LLLA).
- *Partial information:* BS does not know the cost of playing any action except the current action.
    - Binary log-linear learning algorithm (BLLLA).

Best Response Algorithm

1: **Initialisation:** Arbitrary set CRE bias c_i $\forall i \in \mathcal{S}$.
2: **While** $t \geq 1$ **do**
3: Randomly select a BS i.
4: Select its CRE $c_i(t)$ from B_i

$$B_i\left(c_{-i}\right) = arg\ min_{c_i} U_i\left(c_i, c_{-i}\right). \tag{10}$$

5: All the other BSs must repeat their previous actions, i.e., $c_{-i}(t) = c_{-i}(t-1)$.

Theorem ([Monderer and Shapley, 1996])

For a potential game, BR converges to a Nash equilibrium but not necessary the optimum.

Log-linear Learning Algorithm

1: **Initialisation:** Arbitrary set CRE bias c_i $\forall i \in \mathcal{S}$.
2: Set parameter τ.
3: **While** $t \geq 1$ **do**
4: Randomly select a BS i.
5: Select its CRE $c_i(t)$ from X_i with probability

$$p_i^{c_i}(t) = \frac{\exp\left(\frac{1}{\tau} U_i\left(c_i, c_{-i}(t-1)\right)\right)}{\sum_{c_i' \in X_i} \exp\left(\frac{1}{\tau} U_i\left(c_i', c_{-i}(t-1)\right)\right)}. \tag{11}$$

6: All the other players must repeat their previous actions, i.e., $c_{-i}(t) = c_{-i}(t-1)$.

Theorem ([Marden and Shamma, 2012])

For an exact potential game, LLLA converges to the optimal Nash equilibrium.

Binary Log-linear Learning Algorithm

1: **Initialisation:** Arbitrary set CRE bias c_i $\forall i \in \mathcal{S}$.
2: Set parameter τ.
3: **While** $t \geq 1$ **do**
4: Randomly select a BS i.
5: Try an action $\hat{c}_i \in X_i$ uniformly and observe its utility.
6: Play the action $c_i(t) \in \{c_i(t-1), \hat{c}_i\}$ as given below.

$$c_i(t) = \begin{cases} c_i(t-1), & \text{w.p. } \dfrac{e^{\frac{1}{\tau}U_i(\bar{c}(t-1))}}{e^{\frac{1}{\tau}U_i(\bar{c}(t-1))} + e^{\frac{1}{\tau}U_i(\hat{c}_i, c_{-i}(t-1))}}, \\ \hat{c}_i, & \text{w.p. } \dfrac{e^{\frac{1}{\tau}U_i(\hat{c}_i, c_{-i}(t-1))}}{e^{\frac{1}{\tau}U_i(\bar{c}(t-1))} + e^{\frac{1}{\tau}U_i(\hat{c}_i, c_{-i}(t-1))}}. \end{cases} \quad (12)$$

7: All the other players must repeat their previous actions, i.e., $c_{-i}(t) = c_{-i}(t-1)$.

Theorem ([Marden and Shamma, 2012])

For an exact potential game, BLLLA converges to the optimal Nash equilibrium.

Effect of time-varying neighbours

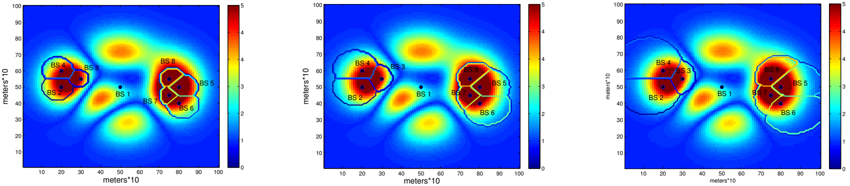
- BSs need to know their neighbours to calculate WLU
- Not an issue if CREs are fixed (cf 3GPP ANR)
- If the neighbor set is changing, we loose the potential game structure
- We can go around as follows:
  - Start algos with any neighbor set
  - During the learning process, update neighbor set when new neighbors are discovered
- As the CRE's set is finite, all sets become constant after some time
- From this instant, the game becomes potential game

Simulation parameters

Table : Simulation parameters.

Parameter	**Variable**	**Value**
Number of BSs	N_s	8
Transmit power of macro BS	P_{macro}	46 dBm
Transmit power of small BS	P_{small}	24 dBm
Average file size	$\frac{1}{\mu}$	0.5 Mbytes
Average traffic load density	$\frac{\lambda}{\mu}$	64 bits$/s/m^2$
System bandwidth	W	20 MHz
Noise power	N	-174+10log(W) dBm
Minimum SINR	γ_{min}	-10 dB
Path-loss exponent	η	3.5
CRE bias set	c_i	$\{1, 1.1, 1.2, \ldots, 16\}$

Optimal Coverage Regions

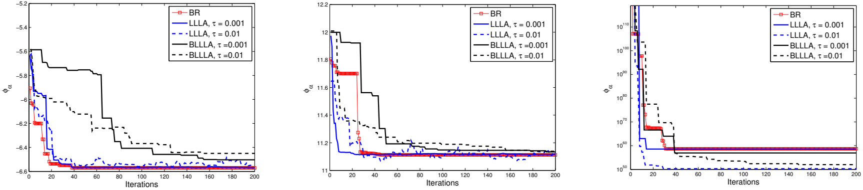


(a) ($\alpha = 0$) Rate-optimal. **(b)** ($\alpha = 2$)Delay-optimal. **(c)** ($\alpha = 200$) Min-max.

Table : Comparison of optimal CRE, optimal loads of BSs for different α.

	$\alpha = 0$		$\alpha = 2$		$\alpha \to \infty$	
BS i	c_i^*	ρ_i^* %	c_i^*	ρ_i^* %	c_i^*	ρ_i^* %
1	1	92	1	62	1	42
2	1.1	9	3.1	20	16	51
3	1	4	3.6	11	16	21
4	1	7	2.8	17	14.8	49
5	1.1	12	3.4	23	7.7	42
6	1.1	8	3.4	20	7.7	41
7	1.1	5	3.5	12	16	25
8	1	6	3.2	18	7.2	42

Convergence of Learning Algorithms



(a) ($\alpha = 0$) Rate-optimal. **(b)** ($\alpha = 2$)Delay-optimal. **(c)** ($\alpha = 200$) Min-max.

Figure : Convergence of BR, LLLA, and BLLLA.

- BR may not converge to the optimal NE.
- For smaller τ, LLLA is similar to BR.
- Partial information is sufficient for implementation.

Conclusion

- Considered a non-convex $\alpha-$fairness objective function.
- ($\alpha = 0$): Rate-optimal, ($\alpha = 1$): proportional fair, ($\alpha = 2$): delay-optimal, and ($\alpha \to \infty$): min-max load policy.
- Formulated CRE selection game as potential game.
- BR, LLLA, and BLLLA are used to achieve Nash equilibrium.
- BR does not necessary converge to optimal Nash equilibrium.
- Partial information is sufficient for practical implementation.

Future Work

- Including shadow fading: all stations are potentially neighbors of all stations, we loose the distributed aspect of the algos.
- Including ABS at the macro: not a big deal, but we have to be careful with outage probability and scheduling policy
- Noisy estimation of the BSs cost function: BSs may have an imperfect knowledge of their own load

Bibliography I

Kim, H., de Veciana, G., Yang, X., and Venkatachalam, M. (2012).
Distributed alpha-optimal user association and cell load balancing in wireless networks.
IEEE/ACM Trans. Netw., 20(1):177–190.

Marden, J. R. and Shamma, J. S. (2012).
Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation.
Games and Economic Behavior, 75(2):788–808.

Monderer, D. and Shapley, L. S. (1996).
Potential Games.
Games and Economic Behaviour, 14(1):124–143.