



HAL
open science

Resource utilization techniques in distributed networks with limited information

Ahmed Farhan Hanif

► **To cite this version:**

Ahmed Farhan Hanif. Resource utilization techniques in distributed networks with limited information. Networking and Internet Architecture [cs.NI]. Institut National des Télécommunications, 2014. English. NNT : 2014TELE0011 . tel-01149690

HAL Id: tel-01149690

<https://theses.hal.science/tel-01149690>

Submitted on 7 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE
CURIE

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Ahmed Farhan HANIF

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Sujet de la thèse :

Resource Utilization Techniques in Distributed Networks with Limited Information

Soutenue le : 7 Mai, 2014.

devant le jury composé de :

M. Luis MUÑOZ	Rapporteur	University of Cantabria, Spain
M. Rahim TAFAZOLLI	Rapporteur	University of Surrey, UK
M. Pierre SENS	Examineur (Président)	Université of Paris 6, France
M. Pascal BIANCHI	Examineur	Telecom ParisTech, France
Ms. Nidhi HEGDE	Examineur	Technicolor Labs, France
M. Mohamad ASSAAD	Co-encadrant	Supelec, France
M. Hamidou TEMBINE	Co-encadrant	KAUST, Saudi Arabia
M. Djamal ZEGHLACHE	Directeur de thèse	Telecom SudParis, France

Thèse n° 2014TELE0011

blank page



JOINT THESIS BETWEEN TELECOM SUDPARIS AND UNIVERSITY OF PARIS 6 (UPMC)

Doctoral School : Informatique, Télécommunications et Electronique de Paris

Presented by

Ahmed Farhan HANIF

For the degree of

DOCTEUR DE TELECOM SUDPARIS

Thesis Title :

Resource Utilization Techniques in Distributed Networks with Limited Information

Defense Date : 7 May, 2014.

Jury Members :

M. Luis MUÑOZ	Reporter	University of Cantabria, Spain
M. Rahim TAFAZOLLI	Reporter	University of Surrey, UK
M. Pierre SENS	Examiner (President)	Université of Paris 6, France
M. Pascal BIANCHI	Examiner	Telecom ParisTech, France
Ms. Nidhi HEGDE	Examiner	Technicolor Labs, France
M. Mohamad ASSAAD	Advisor	Supelec, France
M. Hamidou TEMBINE	Advisor	KAUST, Saudi Arabia
M. Djamal ZEGHLACHE	Director of thesis	Telecom SudParis, France

Thèse n° 2014TELE0011

blank page

Acknowledgements

I would like to express my deepest gratitude to Mohamad Assaad, Hamidou Tembine and Djamal Zeghlache for their guidance, support and constructive feedback throughout my PhD. Under their tutelage, I was well supervised and given the freedom to work on problems of my interest. They were instrumental in motivating me and giving me direction whenever it was required.

I would like to thank the jury members for allocating their precious time to participate in my defense.

I would also like to thank the people who were helpful at different stages of my work.

- A special thanks to Valerie Mateus of RS2M Lab in Telecom SudParis and Catherine Magnet of Telecommunications Department at Supelec for the logistical support regarding various official matters.
- I would like to thank all my friends from Supelec and Telecom SudParis, among them particularly Mereyem, Daniel, Shoaib, Li-Chuan, Jakob, Stefoan, Subhash, Ayaz, Marwen for their company.

Finally, I would like to thank my family, especially my beloved wife Kiran, for her continued support during my PhD.

Ahmed Farhan Hanif, May 2014.

blank page

Abstract

As systems are becoming larger, it is becoming difficult to optimize them in a centralized manner due to insufficient backhaul connectivity and dynamical systems behavior. In this thesis, we tackle the above problem by developing a distributed strategic learning framework for seeking Nash equilibria under state-dependent payoff functions. We develop a discrete time stochastic learning using sinus perturbation with the realistic assumption, that each node only has a numerical realization of the payoff at each time. We examine the convergence of our discrete time algorithm to a limiting trajectory defined by an ordinary differential equation (ODE). Finally, we conduct a stability analysis and apply the proposed scheme in a generic wireless networks. We also provide the application of these algorithms to real world resource utilization problems in wireless. Our proposed algorithm is applied to the following distributed optimization problems in wireless domain. Power control, beamforming and Bayesian density tracking in the interference channel.

We also consider resource sharing problems in large scale networks (e.g. cloud networks) with a generalized fair payoff function. We formulate the problem as a strategic decision-making problem (i.e. a game). We examine the resource sharing game with finite and infinite number of players. Exploiting the aggregate structure of the payoff functions, we show that, the Nash equilibrium is not an evolutionarily stable strategy in the finite regime. Then, we introduce a myopic mean-field response where each player implements a mean-field-taking strategy. We show that such a mean-field-taking strategy is evolutionarily stable in both finite and infinite regime. We provide closed form expression of the optimal pricing that gives an efficient resource sharing policy. As the number of active players grows without bound, we show that the equilibrium strategy converges to a mean-field equilibrium and the optimal prices for resources converge to the optimal price of the mean-field game. Then, we address the demand satisfaction

problem for which a necessary and sufficiency condition for satisfactory solutions is provided.

Resumé

Dans ce travail, notre contribution est double. Nous développons un cadre d'apprentissage stochastique distribué pour la recherche des équilibres de Nash dans le cas de fonctions de paiement dépendantes d'un état. La plupart des travaux existants supposent qu'une expression analytique de la récompense est disponible au niveau des nœuds. Nous considérons ici une hypothèse réaliste où les nœuds ont seulement une réalisation quantifiée de la récompense à chaque instant et développons un modèle stochastique d'apprentissage à temps discret utilisant une perturbation en sinus. Nous examinons la convergence de notre algorithme en temps discret pour une trajectoire limite définie par une équation différentielle ordinaire (ODE). Ensuite, nous effectuons une analyse de la stabilité et appliquons le schéma proposé dans un problème de commande de puissance générique dans les réseaux sans fil. Nous avons également élaboré un cadre de partage de ressources distribuées pour les réseaux –cloud– en nuage. Nous étudions la stabilité de l'évolution de l'équilibre de Nash en fonction du nombre d'utilisateurs. Dans ce scénario, nous considérons également le comportement des utilisateurs sociaux. Enfin nous avons également examiné un problème de satisfaction de la demande où chaque utilisateur a une demande propre à lui qui doit être satisfaite.

Introduction la recherche du Nash

L'optimisation distribuée a bénéficié un intérêt croissant au cours des dernières années en raison de son application dans des domaines tels que les réseaux sans fil, l'estimation distribuée, etc. Le problème principal se compose d'un réseau de n nœuds ou agents en interaction où chacun a une récompense ou une fonction d'utilité à maximiser. La décision de chaque nœud a un impact sur la fonction de récompense des autres nœuds, ce qui rend le problème difficile en général.

Même si le problème d’optimisation distribuée, considéré dans ce travail, et l’approche développée sont généraux et peuvent être utilisés dans de nombreux domaines d’application, nous nous limitons au problème de commande de puissance dans les réseaux sans fil afin de mieux illustrer notre contribution. La généralisation des résultats obtenus à d’autres applications n’est pas difficile en soi.

Recherche d’extremum

Krstic *et.al.* au cours des dernières années ont énormément contribué au domaine de la recherche d’extremas non-basée sur un modèle. Dans [FKB12a], les auteurs proposent un algorithme cherchant l’équilibre de Nash pour les jeux avec des espaces d’action continus. Ils proposent un algorithme d’apprentissage entièrement distribué qui nécessite seulement une mesure de la valeur numérique de la récompense. Leur algorithme est fondé sur la perturbation en sinus (perturbation dite déterministe au lieu d’une perturbation stochastique) de la fonction de paiement en temps continu. Toutefois, le schéma d’apprentissage à temps discret avec des perturbations en sinus n’est pas examiné dans [FKB12a]. Dans [SS10], l’algorithme de recherche d’extremas avec perturbations sinusoïdales pour des systèmes non-basés sur des modèles a été étendu et modifié pour le cas de mesures bruitées i.i.d et de perturbations en sinus évanouissantes, la convergence presque sûre à l’équilibre est prouvée. La recherche d’extremas basée sur la perturbation en *sinus* pour *des mesure de bruit indépendantes de l’état* est présentée dans [SJS12].

Dans ce travail, nous étendons le travail dans [FKB12b] au cas de fonctions de paiement textit dépendantes d’un état stochastique, et utilisons des perturbations déterministes pour recherche de l’équilibre de Nash. On peut facilement voir la différence entre ce travail et les travaux existants [SS10, SJS12]. Dans ces travaux, le bruit η_j associé à la mesure est additif et i.i.d , ce qui n’est pas nécessairement vrai dans la pratique notamment dans le domaine de l’ingénierie, où le bruit est en général corrélé dans le temps. Dans notre cas, nous considérons une fonction stochastique de gain dépendantz de l’état et notre problème peut être écrit sous une forme de Robbins-Monro avec un bruit Markov (corrélé) donné par $\eta_j = r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}}[r_j(\mathbf{S}, \mathbf{a})]$ (Cela deviendra plus clair dans les sections suivantes), c’est à dire que le bruit associé est stochastique dépendant de l’état et de l’action , ce qui est différent du cas du bruit i.i.d additif supposé dans

[SS10, SJS12].

Contribution

Dans ce travail, nous proposons un *temps discret* algorithme d'apprentissage, en utilisant la perturbation en sinus *continue* pour des jeux d'action où chaque noeud ne dispose que d'une réalisation numérique de leur propre récompense à chaque instant. Nous étendons donc la méthode classique de recherche de l'équilibre de Nash avec perturbation en sinus [FKB12b] au cas du temps discret et des fonctions de paiement dépendantes de l'état stochastique. Nous montrons que notre algorithme converge localement à un état d'équilibre de Nash indépendant de l'état dans le théorème 1 dans [HTAZ13d] pour un pas évanouissant et nous bornons l'erreur correspondante dans le théorème 2 dans [HTAZ13d] pour un pas fixe. Notez que puisque les fonctions de paiement ne sont pas nécessairement concaves, trouver un optimum global en un temps raisonnable peut être difficile en général, même dans le cas déterministe (état fixe) et avec une expression analytique de la fonction de gain. Nous montrons aussi le temps de convergence pour le cadre de sinus dans le corollaire 1 dans [HTAZ13d].

Les publications suivantes ont été soumises ou ont été publiées dans des revues et / ou conférences.

1. La principale contribution de la recherche d'extremum et notamment les preuves sont présentés dans [HTAZ13d].
2. Nous avons appliqué cette technique de recherche d'extremum pour les systèmes sans fil dans les scénarios suivants.
 - (a) Apprentissage distribué stochastique pour le contrôle de puissance continue dans les réseaux sans fil a été présenté dans [HTAZ13a].
 - (b) Une commande de puissance distribuée dans les cellules femto bas" sur une poursuite de densité bayésienne a été développée dans [HTAZ12c].
 - (c) Dans [HTAZ13b], nous avons développé une stratégie de beamforming d'émission distribuée avec 1 bit de rétroaction pour les canaux LoS-MISO.

Définition du problème et algorithme suggéré

Nous considérons le problème du contrôle de puissance dans les réseaux sans fil qui est composé de paires émetteur-récepteur utilisant tous la même fréquence et ainsi générant des interférences les uns sur les autres. Chaque paire émetteur - récepteur possède donc sa propre fonction de récompense / utilité qui dépend nécessairement de l'interférence exercée par les autres paires / nœuds. Comme le gain de canal sans fil et l'interférence varient dans le temps, l'objectif est d'optimiser nécessairement un long terme (par exemple moyenne) des fonctions de récompense de tous les nœuds. La fonction de récompense de l'utilisateur i au temps k est notée $r_i(\mathbf{S}_k, \mathbf{a}_k)$ où $\mathbf{S}_k := [h_k(i, j)]$ représente une matrice $n \times n$ contenant les coefficients de canal au temps k , $h_k(i, j)$ représente le coefficient de canal entre l'émetteur i et le récepteur j (où $(i, j) \in [1, \dots, n]^2$) et \mathbf{a}_k représente le vecteur contenant la puissance de transmission de n nœuds. La technique la plus couramment utilisée pour obtenir un maximum local des fonctions de récompense des nœuds est la méthode de de gradient croissants / décroissants. Supposons qu'il y ait n nœuds transmetteurs récepteurs chacun avec une fonction de récompense représentée par $r_i(\mathbf{S}_k, \mathbf{a}_k)$ qui est utilisée pour formuler les problèmes d'optimisation suivants:

$$\max_{a_i \geq 0} \mathbb{E}_{\mathbf{S}} r_i(\mathbf{S}, \mathbf{a}) \quad \forall i \in \{1, \dots, n\} =: \mathcal{N} \quad (1)$$

où $\mathbb{E}_{\mathbf{S}}$ désigne l'opérateur espérance par rapport à \mathbf{S} .

En supposant que nous avons accès à la récompense à chaque instant k , mais l'expression analytique de $r_i(\mathbf{S}_k, \mathbf{a}_k)$ est inconnue pour l'utilisateur i . une solution au problème ci-dessus est un équilibre indépendant de l'Etat dans le sens qu' aucun émetteur n'a intérêt à changer sa puissance d'émission lorsque les autres émetteurs conservent leur choix. Il est bien connu que les équilibres peuvent être différents des optimum global, l'écart entre l'équilibre et le pire maximum global est capturé par le soi-disant *prix de l'anarchisme* . Ainsi la solution obtenue par notre méthode peut être sous-optimale par rapport à la maximisation de la somme de toutes les récompenses. Nous étudions la stabilité locale de l'algorithme stochastique.

Puisque la structure des fonctions de paiement est inconnue, les gradients ne peuvent pas être calculés. Ainsi, chaque réseau n'a pas besoin d'être au courant de la présence de ces autres réseaux. La seule information dont chaque émetteur

a besoin est la valeur numérique $r_{i,k}$. Les fonctions de paiement sous-jacentes ne sont pas supposées être concaves / convexes. Ce genre de système est aussi appelé système à modèle libre. Nous supposons seulement que les fonctions sont Lipschitz et qu'un point Nash équilibre existe.

Algorithme d'apprentissage

Supposons que chaque utilisateur i est en mesure d'observer une valeur numérique $r_{i,k}$ de la fonction $r_i(\mathbf{S}_k, \mathbf{a}_k)$ au temps k , où $a_{i,k}$ est la puissance d'émission de l'émetteur i au temps k . $\hat{a}_{i,k}$ est une variable intermédiaire. a_i , Ω_i , ϕ_i représentent la fréquence d'amplitude et la phase du signal de perturbation en sinus, $r_{i,k+1}$ représente la récompense à l'instant $k+1$.

A chaque instant k , chaque émetteur met à jour sa puissance $a_{i,k}$, en ajoutant la perturbation en sinus à la variable intermédiaire $\hat{a}_{i,k}$ en utilisant l'équation 2.31, et reprend la transmission en utilisant $a_{i,k}$. Ensuite, chaque émetteur reçoit une réalisation de la récompense $r_{i,k+1}$ de son récepteur correspondant au temps $k+1$ qui est utilisée pour calculer $\hat{a}_{i,k+1}$ en utilisant l'équation 3. La puissance $a_{i,k+1}$ est ensuite mise à jour en utilisant l'équation 2.31. Cette procédure est répétée pour tout au long de la fenêtre de transmission T . L'algorithme est à temps discret et est donnée par

$$a_{i,k} = \hat{a}_{i,k} + a_i \sin(\Omega_i \hat{k} + \phi_i) \quad (2)$$

$$\hat{a}_{i,k+1} = \hat{a}_{i,k} + \lambda_k l_i a_i \sin(\Omega_i \hat{k} + \phi_i) r_{i,k+1} \quad (3)$$

où $\hat{k} := \sum_{k'=1}^k \lambda_{k'}$. $\Omega_i \neq \Omega_j$, $\Omega_j + \Omega_i \neq \Omega_{j'}$ et $\Omega_j \forall i$ devraient être choisis assez faibles. Par exemple, $\lambda_k = \frac{1}{k+1}$. $\phi_i \in [0, 2\pi] \forall i$, $k \in \mathbb{Z}$

Chaque utilisateur i , initialiser $\hat{a}_{i,0}$ et transmettre
 Répétez
 Calcul puissance selon l'équation (2.31)
 Observer $r_{i,k}$
 mise à jour $\hat{a}_{i,k+1}$ en utilisant l'équation 3
 jusqu'à horizon T

Algorithm 1: algorithme d'apprentissage distribué

Introduction au partage des ressources dans le Cloud

Les solutions de partage de ressources sont très importantes pour les centres de données comme car il est mis-en œuvre à différents niveaux du réseau de nuage. Le problème de partage de ressources peut être formulé comme un problème de décision stratégique entre les clients. Dans une telle situation, mais dans un contexte différent, Tullock (1980, [Tul80]) a proposé un schéma de la théorie des jeux à récompense probabiliste it recherche de rente. Il a remarqué plus tôt que la perte sociale associée à la recherche de rente dépasse un certain seuil. Du point de vue de la mise en réseau des nuages, ce qui signifie que beaucoup de ressources peuvent être perdues si l'utilisateur nuage envisage une location économique. Dans ce travail, nous étudions un modèle stylisé de partage des ressources de manière distribuée et équitable. Nous allons montrer que l'efficacité peut être considérablement améliorée pour les grands réseaux de cloud computing en utilisant une conception appropriée de prix. Nous examinons à la fois la stabilité et la réaction myope utilisant des stratégies à champ moyen[AFA05]. Les jeux de champ moyen sont connus pour être bien adaptés pour les systèmes finis (mais grands) et infinis. Contrairement aux autres outils classiques pour les systèmes à grande échelle, l'approche de champ moyen dynamique intègre les dynamiques qui permettent la gestion de la demande en ligne (pay-as-you-use régime). Elle permet l'optimisation, le contrôle et la conception dynamique de mécanisme lorsque le nombre d'utilisateurs actifs varie et étudie la nature interactive des réseaux de nuage, conduisant à des problèmes de prise de décisions stratégiques. Des études récentes dans les réseaux de cloud ont déjà adopté un régime de champ moyen. Par exemple, dans [PB11b], les auteurs considèrent cloud public de plus de cent mille serveurs et de nombreux nuages privés en plus de l'échelle du réseau traditionnel.

L'une des questions fondamentales est de connaître l'écart d'erreur ou erreur d'approximation du modèle de champ moyen. En d'autres termes, la précision de l'approximation fournie par le modèle de champ moyen dans un réseau de nuages avec nombre fini (mais grand) d'utilisateurs.

Nous fournissons une réponse complète à la question ci-dessus dans ce travail.

Contribution

Bien que nous considérons ici les réseaux de cloud computing, notre analyse qui en résulte est plus générale et est applicable à des réseaux similaires de partage

de ressources. Les contributions de ce travail sont listées ci-dessous.

- On considère des jeux de partage des ressources en nuage avec des espaces d'action continus où chaque utilisateur tente de maximiser son propre gain. Dans le régime fini (nombre limité d'utilisateurs), nous fournissons les expressions analytiques de trois importants concepts de solution en théorie des jeux: équilibre de Nash (NE), réponse de champ moyen à population finie (F - MFR) et stratégies évolutives stables à population finie (F - ESS).
- Dans le régime fini, nous fournissons alors un algorithme itératif basé sur Ishikawa qui converge vers chacun des trois concepts d'équilibre ci-dessus et discutons ses temps de convergence.
- On considère alors le régime infini, car dans de nombreux réseaux de nuages le nombre d'utilisateurs devrait être très élevé. Nous fournissons une expression analytique de l'équilibre du champ moyen et montrons que l'équilibre du champ moyen est une stratégie évolutivement stable, c'est à dire qu'il ne peut pas être envahi par une petite fraction des déviants. Nous montrons que l'équilibre du champ moyen est stable évolutif pour toute fraction de déviants strictement inférieure à 100% ce qui rend notre résultat encore plus fort. Nous montrons également que l'équilibre de Nash est stable dans le régime infini.
- Ensuite, nous nous concentrons sur l'algorithme d'apprentissage distribué pour le régime infini. Nous proposons un système d'apprentissage (mais toujours avec moins d'informations) basé sur un modèle pour les jeux avec l'espace de l'action continue et à grand nombre de joueurs. Chaque joueur va mettre à jour sa stratégie d'apprentissage basée sur une durée d'agrégation, qui est la moyenne d'une fonction croissante de l'action des autres joueurs.
- La dernière contribution de ce papier est l'étude de concept de jeu satisfaisant. Chaque utilisateur du nuage a une demande de capacité qui doit être satisfaite n'importe quand n'importe où (au lieu d'essayer de maximiser un profit). Dans le régime fini, nous fournissons une expression analytique de la solution satisfaisante et développons un algorithme distribué qui est capable de l'atteindre.

Pour plus de détails sur les contributions ci-dessus et des résultats numériques veuillez vous référer à [HTAZ13c, HTAZ12a].

System Model

Nous considérons le partage des ressources jeu avec un nombre fini (mais arbitraire) de clients dans un réseau de nuage. On note un tel jeu par \mathcal{G}_n , où n est le nombre de clients. L'espace d'action de chaque utilisateur est $\mathcal{A} = \mathbb{R}_+$ qui est un ensemble convexe, c'est-à-dire chaque joueur j choisit une action a_j qui appartient à l'ensemble \mathcal{A} . l'action peut représenter une certaine demande. Toutes les actions déterminent ensemble un résultat. Soit p_n le prix unitaire de l'utilisation des ressources de cloud par les clients.

Ensuite, le gain de joueur j est donné par:

$$r_j(a_1, \dots, a_n) = c_n \frac{h(a_j)}{\sum_{i=1}^n h(a_i)} - p_n a_j, \quad (4)$$

si $\sum_{i=1}^n h(a_i) > 0$ et zéro autrement. La structure de la fonction récompense $R_j(a_1, \dots, a_n)$ pour l'utilisateur j montre qu'il s'agit d'un pourcentage de la capacité allouée moins le coût d'utilisation de cette capacité. Ici, c_n représente la valeur des ressources disponibles (qui peuvent être considérées comme la capacité du nuage), h est une fonction positive et décroissante avec $h(0) = 0$. Nous fixons la fonction h être x^α where $\alpha > 0$ dénote un certain indice de rendement.

Conclusion

Nous avons étudié les jeux de partage des ressources dans les réseaux de cloud computing, mais notre analyse est applicable à d'autres scénarios de partage des ressources. Pour le cas fini, nous avons prouvé que le point d'équilibre Nash est instable pour notre fonction de paiement et avons identifié deux stratégies stables à savoir F - ESS et F - MFR. Nous avons aussi développé des algorithmes qui sont capables de converger respectivement aux stratégies mentionnées dans la section précédente. Pour les systèmes à grande échelle, nous avons développé un schéma de jeu de champ moyen et montré que l'équilibre du champ moyen est stable. Nous développons ensuite un algorithme itératif distribué qui converge vers l'équilibre du champ moyen avec un ordre de convergence de quatre. En outre, nous avons donné des conditions nécessaires et suffisantes pour l'existence de solution satisfaisante où chaque utilisateur est satisfait n'importe

où n'importe quand. Pour plus de détails sur les contributions ci-dessus et les résultats numériques veuillez vous référer à [[HTAZ13c](#), [HTAZ12a](#)].

Contents

1	Introduction	1
1.1	Motivation and Challenges	1
1.1.1	Wireless Resource Utilization	2
1.1.2	Cloud Networks Resource Sharing	5
1.2	Outline and Contributions	7
1.2.1	Extremum Seeking Algorithm	7
1.2.2	Distributed Resource Sharing	9
1.2.3	Publication List	11
2	Model Free Nash Seeking	12
2.1	Introduction	12
2.2	Problem Formulation and Proposed Algorithm	17
2.2.1	Distributed learning algorithm	18
2.2.2	Interpretation of the proposed algorithm	19
2.3	Main results	20
2.4	Numerical Example A: Wireless Interference	25
2.5	Numerical Example B: Distributed Beamforming	29
2.6	Numerical Example C: Nash Bayesian Estimation	33
3	Resource Sharing in Cloud Networks	46
3.1	Introduction	46
3.1.1	Overview	49
3.1.2	Contribution	50
3.1.3	Structure	52
3.2	Model	52
3.2.1	Resource sharing game \mathcal{G}_n with finite number of clients	53
3.2.2	Resource sharing game \mathcal{G}_∞ with infinite number of clients	55

3.2.3	Demand satisfaction game \mathcal{G}_s	56
3.3	Equilibrium Analysis	57
3.3.1	Equilibrium for \mathcal{G}_s finite number of clients	57
3.3.2	Equilibrium for \mathcal{G}_∞ infinite number of clients	61
3.3.3	Equilibrium for \mathcal{G}_s Demand Satisfaction	63
3.4	Algorithms	64
3.4.1	Iterative Learning Algorithm in the Finite Regime	64
3.4.2	Iterative Learning Algorithm in the Infinite Regime	66
3.4.3	Iterative Learning Algorithm for satisfactory solution	69
3.5	Numerical Investigation	70
3.5.1	Finite user scenario	70
3.5.2	Infinite user scenario	70
3.5.3	Heterogeneous pricing scenario	71
4	Conclusion and Future Work	77
4.1	Extremum Seeking	77
4.2	Resource Sharing Games in Cloud Networks	78
4.3	Future Work	79
A	Proof of convergence in Extremum Seeking	81
A.1	Variable Step Size: Proof of Theorem 1	81
A.2	Fixed Step Size: Proof of Theorem 2	86
A.3	Stability analysis, Proof of Theorem 3	89
A.4	Conditions for Example A	93
B	Proofs in Mean Field Resource Sharing	94

Acronyms

a.s. almost sure

CDMA code division multiple access

CMP Chip Microprocessor

discrete ODE the learning algorithm

DoT direction of transmission

F - ESS stratégies évolutives stables à population finie

F - MFR réponse de champ moyen à population finie

F-ESS finite evolutionary stable strategy

F-MFR last level cache

FAP femto access point

FC femtocell

IaaS Infrastructure as a Service

LLC last level cache

LoS line of sight

LoS line of sight

MC macrocell

MIMO multiple input multiple output

MISO multiple input single output

NE Nash equilibrium

NE équilibre de Nash

ODE ordinary differential equation

OFDMA orthogonal frequency division multiple access

PaaS Platform as a Service

QoS quality of service

RaaS Resource-as-a-Service

SaaS Software as a Service

SINR signal to interference and noise ratio

SISO single input single output

SLA services level agreement

ULA uniform linear array

VM virtual machine

XaaS anything as a service

Definition 1	Nash Equilibrium (state-independent)	17
Definition 2	Nash Equilibrium (state-dependent)	18
Definition 3	ϵ -Nash equilibrium payoff point	24
Definition 4	ϵ -close strategy Nash equilibrium point	24
Definition 5	Demand Profile	57
Definition 6	Symmetric Game	58
Definition 7	F-ESS	60
Definition 8	F-MFR	61
Definition 9	Mean-field (Nash) equilibrium	62
Definition 10	ESS	62
Definition 11	<i>Invasion barrier</i> $\bar{\epsilon}_m$	63
Definition 12	66
Definition 13	Convergence time	68

Chapter 1

Introduction

In this chapter we present a general description of problems in future networks. In particular we identify and focus on the technological challenges behind these problems. Finally we present the main contribution in this thesis. References are given throughout the work, where interested readers may find detailed information if necessary.

1.1 Motivation and Challenges

The main motivation for this thesis comes from the challenges that exists in the design and development of future networks. Future networks will support an ever greater number of subscribers and higher data rates per subscriber than existing networks. At the same time these networks will need to utilize the available resources more efficiently, but above all they will have to consume low power. Future network is a blanket term which includes all types of wired and wireless communications networks, including the concepts of ‘cloud networks’, ‘internet of things’ and ‘wireless sensor network’. There is a great technological gap that needs to be bridged in-order to satisfy the performance claims hinted above.

Most of the existing network designs are based on a centralized controller which is responsible for all resource management and control requests but such models are non-scalable to higher number of users as it entails a very high signaling on the backhaul link.

Another limitation even at the small scale is when there are several non cooperating users, for example in cognitive radio we have primary and secondary users using the same spectrum at the same time and interfering with one another.

In such cases the system is inherently distributed and hence the problem cannot be solved in a centralized manner.

To overcome these challenges there is a growing consensus among network designers to make future networks decentralized where each node is autonomous and is able to make intelligent decisions about resource utilization. This decision or action has to be updated at each time as the demand of the network is constantly changing i.e. dynamic.

This type of approach has certain design challenges which are listed as follows:

- How to implement a distributed optimization when there is only limited information in feedback?
- How to ensure that the distributed optimization problem converges to a local maxima?

The answer to both of the above question is dependent on the structure of the reward function and the available information for updates. There can be several practical limitation associated with the reward function which are listed below. In some scenarios, the reward function $r_j(\cdot)$ associated with each node is not known, or one or more dependent variables of the reward functions are not known. The reward function could be known but non differentiable. The node may not have the computing power necessary to perform the computation required for optimization. The state of the system could be time varying. All these above limitations encourage designers to look at more clever techniques to deal with such challenges. One way to deal with such limitations is by using a ‘model free’ approach to optimization. We will discuss these approaches later in this chapter.

1.1.1 Wireless Resource Utilization

The wireless resources are an expensive and limited commodity. There are several methods that have been developed over the year to utilize the wireless resources in a much more efficient ways. For example using efficient multiple access techniques such as code division multiple access (CDMA) and orthogonal frequency division multiple access (OFDMA) to divide the spectrum between a greater number of subscribers, installing multiple antennas per subscriber i.e. multiple input multiple output (MIMO) systems to improve capacity or performance, using more efficient coding techniques to better protect against random-

ness in the channel. Having faster more robust routing protocols which reduce latency by choosing the shortest path through the network. As the demand for data rate is increasing wireless infrastructure is becoming overloaded with more and more users. This surge in the number of users is putting higher demand on the backhaul to resolve the interference in intelligent ways. This kind of scaling is not practical as the backhaul links has a limited bandwidth.

Motivation for decentralized system: One logical progression is to transition from a centralized to a distributed architecture. In wireless communications there are several scenarios which are inherently distributed i.e. for example an ad-hoc network which does not rely on a pre existing infrastructure. Cognitive radio network is another example where there are primary and secondary users. The primary users represent the pre-existing infrastructure and operates in a licences frequency spectrum. The secondary users are allowed to use the spectrum only when the primary users are not using it. This type of network is a distributed network as there is no central controller between the primary and the secondary users. Another reason for moving towards a decentralized system is that the delay associated with a centralized system which may cause the information to become outdated by the time it has been shared and used by the central controller. Hence there is greater demand to develop fully distributed techniques which are able to achieve performance similar to a centralized systems. Distributed optimization has been receiving an increasing interest over the past years due to its ability of use in many application fields (wireless networks, distributed estimation, etc.). The main problem consists in a network of n interacting nodes or agents where each one has a reward or utility function to maximize. The decision of each node has an impact of the reward function of the other nodes, which makes the problem challenging in general. Different approaches, mainly based on gradient descent/ascent method, have been developed to achieve a local optimum (or global optimum in some special cases, e.g. concavity of the reward, etc.) of the distributed optimization problem. But in practical systems it may not be possible to implement gradient based method due to lack of knowledge about the reward function.

There are many problem in wireless communication where only a noisy or quantized numerical value of reward is available in feedback at the transmitter. In such cases implementing a distributed optimization approach becomes much more challenging.

Scenarios that we have considered: Following is a brief description of some of the problems in distributed wireless resource utilization that we have considered.

- **Power Control in Interference Channel:** In wireless communications, an interference channel arises when there are more than one transmitters trying to use the same frequency. The problem is composed of transmitter-receiver pairs; all of them use the same frequency and thus generate interference onto each other. It is an important scenario, and one way of managing the interference is by implementing some type of power control scheme. We consider a challenging scenario where the transmitters do not cooperate with one another and only have to do power control based on their local information. Each transmitter-receiver pair has therefore its own reward/utility function that depends necessarily on the interference exerted by the other pairs/nodes. Since the wireless channel is time varying as well as the interference, the objective is necessarily to optimize a long-run (e.g. average) of the reward functions of all the nodes. The reward function of user i at time k is denoted by $r_i(\mathbf{H}_k, \mathbf{p}_k)$ where $\mathbf{H}_k := [h_k(i, j)]$ represents an $n \times n$ matrix containing channel coefficients at time k , $h_k(i, j)$ represents the channel coefficient between transmitter i and receiver j (where $(i, j) \in [1, \dots, n]^2$) and \mathbf{p}_k represents the vector containing transmit powers of n transmit receive nodes.
- **Beamforming in Interference Channel:** In cases when we have multiple transmitting antennas at each transmitter, it is possible to implement beamforming instead of power control which is a more advanced method of steering the signal power toward your respective receiver. This is a very useful technique as it helps us in avoiding interfering with the other receivers and improve our performance by directing our signal in the direction of our respective receiver. But in order to implement beamforming we need to have some knowledge about the position of the receiver with respect to the transmitter. As we assume a case where we only have a limited information i.e. 1-bit in feedback at each time. We attempt to do distributed beamforming using only one bit feedback from the receiver, which is indeed a very limited information for this task. This is what makes it a challenging problem. We consider the case of line of sight (LoS) channel to keep things simple. Each transmitter will adapt its direction of transmission (DoT) using only the 1-bit feedback in such a way that the

probability of successful transmission is maximized.

- **Bayesian Density Tracking:** In wireless communication systems, as the channels are randomly varying, it makes sense to measure the performance of the channel as the probability of ‘signal strength’ greater than a certain threshold. The mathematical models behind these probability function can be complicated and in most cases the dependent variables may be unknown or randomly varying. We attempt to track the evolution the density of the signal strength and use this information to maximize the success probability using the concepts from Bayesian density tracking. There are scenarios where the probability of success maximization can be a good way to analyze and optimize the performance of a randomly varying system. Here we consider an overlapping femto cell and macro cell operating at the same frequency. We only consider the downlink scenario where the femto cell users will receive interference from the macro access point and vice versa. We construct the reward functions in such a way that the femto access point take the success probability of the macro cell users and its own users into consideration when implementing its power control strategies. The femto and macro cells are two distributed systems and there is not cooperation on the backhaul. Femto has knowledge about the performance of its own user and the performance of the macro user which the femto is able to overhear in the feedback control channel as they are both using the same standard. This is again a distributed scenario with limited information in feedback.

We mainly concentrated on distributed scenarios with single input single output (SISO) and multiple input single output (MISO) channels. In all of the above mentioned scenarios we use our ‘model free’ optimization algorithm that we will discuss in the next chapter.

1.1.2 Cloud Networks Resource Sharing

Cloud Networking is a relatively new concept which ensures that computing and storage resources are available to users on demand. Foster et al.(2008) defines the ambiguous cloud as “A large-scale of distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and ser-

vices are delivered on demand to external customers over the Internet” [FZRL08]. The concept of cloud networking has emerged to satisfy the uneven demand of users wanting cheap computing resources ranging from storage to processing. Every aspect of a computer system can essentially be offered as a service to various users. Thanks to advances in virtualization and scheduling techniques it is possible to have multiple instances of a software running on the same physical system. Although there are many types of cloud based resources that can be shared with other users here we only discuss computing and storage resources.

Computing Resources: Consider a scenario where an entity requires high computing resources once every month. Now they can rent computing resources on demand instead of physically owning dedicated computing resources. With the advent of modern operating systems and advances in networking technologies and virtual machine technologies it is now possible to host and multiple instances with limited hardware. Amazon has been among the first companies to convert this into a viable business with the ‘EC2 service’. ‘Compute Engine by Google’ and ‘Windows Azure by Microsoft’ are the other major players in computing resources. These services are mostly used by companies and research organizations.

Storage Resources: Data storage and backup in the cloud is also an important business. Several companies are providing data storage in the cloud as their primary service, and the demand is to have instant and seamless access to the data over cross platform devices which makes such a service very attractive. As most of our data is now digital it necessary to have secure backup of personalized data. These services are used by individuals and companies. Dropbox, Google Drive, Microsoft SkyDrive are some of the major storage services for users.

The resource sharing structure used by most cloud service providers uses CPU-hour consumed [Inc08], which may not be the most efficient, as it doesn’t take into account the complex interactions between the users that are happening indirectly through the shared resource. As different users are sharing the same physical machines, they are interacting with each other through the machines. The service provider is only interested in maximizing his resource utilization i.e. all CPU-hour are being utilized, but from the users standpoint this may not result in best performance if CPU-hours are divided among a large number of users. The tasks of a single user may take a very long time to finish. Hence the reward of each user is not fully captured by the CPU-hour resource sharing

method, the percentage of resources made available to each user should be part of the reward. We consider such type of reward function where users are interested in increasing their percentage based reward function.

Scenarios that we have considered: Following is a brief description of some of the problems in large scale distributed cloud networks that we have considered.

- One of the problems in sharing resources on a large scale is again with the signaling overhead required to manage a dynamically changing demand by all the users. We address this distributed problem using the concepts from game theory and mean field.
- Demand satisfaction is another resource utilization problem, where each user has a specified demand that needs to be met for the user to be satisfied. This is inherently a different type of resource sharing problem.

Please refer to chapter 3 for detailed information about these cases.

1.2 Outline and Contributions

A general system overview which is common to all scenarios considered in this thesis is presented in figure 1.1. Our model assumes a number of nodes interacting with a dynamic environment. The dynamic environment means that the state of the system is changing. The nodes are non-cooperative and they are interacting with each other through the dynamic environment. The action of each node has an impact on the reward of the other nodes and vice versa. This is the basic system mode that we have considered in our thesis.

1.2.1 Extremum Seeking Algorithm

In chapter 2 we develop the concepts associated with extremum seeking for ‘model free’ scenarios (i.e. where the structure of the reward function is not known) we have developed a stable extremum seeking algorithm that is able to converge to Nash equilibrium when initialized in the neighborhood of the Nash equilibrium point. The algorithm ensures tractability which can be a requirement in certain engineering applications. Note that, our convergence is in the sense of Borkar [Bor08] and will be explained later on in the text.

In this work, we propose a *discrete time* learning algorithm, using sinus perturbation, for *continuous* action games where each node has only a numeri-

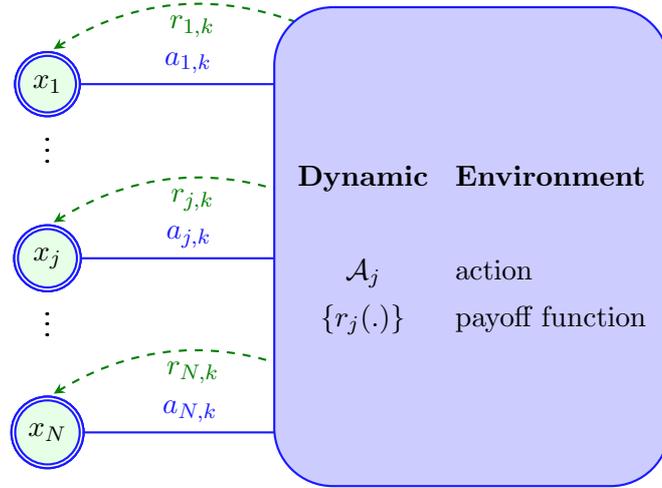


Figure 1.1: Nodes interacting with each other through a dynamic environment

cal realization of own-payoff at each time. We therefore extend the classical Nash Seeking with sinus perturbation method [FKB12b] to the case of discrete time and stochastic state-dependent payoff functions. We prove that our algorithm converges locally to a state independent Nash equilibrium in Theorem 1 in [HTAZ13d] for vanishing step size and provide an error bound in Theorem 2 in [HTAZ13d] for fixed step size. Note that since the payoff function may not necessarily be concave, finding a global optimum in reasonable time can be difficult in general even in deterministic case (fixed state) and known closed-form expression of payoff. We also show the convergence time for the sinus framework in Corollary 1 in [HTAZ13d].

Following publications have been submitted or have appeared in peer reviewed journals and/or conferences.

1. The main contribution of extremum seeking framework including proofs is presented in [HTAZ13d]. A Nash seeking algorithm which is able to find the local minima using just the numerical value of the stochastic state dependent payoff function at each discrete time step. We proved the convergence of our algorithm to a limiting ODE. We have provided as well the error bound for the algorithm and the convergence time to be in a close neighborhood of the Nash equilibrium. A numerical example for a generic wireless network is provided for illustration. The convergence bounds achieved by our method are dependent on the step size and the

perturbation amplitude. We consider a new class of state dependent payoff functions $r_j(\mathbf{S}, \mathbf{a})$ which are inspired from wireless systems applications. But these kind of functions are more general and appear in other application areas. We present an iterative algorithm which is able to perform distributed resource sharing (for same and different prices) for a certain type of reward function. We show the stationary points of our algorithms converge to Nash equilibrium.

2. We have applied this extremum seeking techniques to wireless systems in the following scenarios.
 - (a) Distributed stochastic learning for continuous power control in wireless networks was presented in [HTAZ13a].
 - (b) In [HTAZ13b] we developed a distributed transmit beamforming with 1-bit feedback for LoS-MISO channels.
 - (c) A distributed power control in femto cells using bayesian density tracking was developed in [HTAZ12c].

1.2.2 Distributed Resource Sharing

In chapter 3 we describe distributed resource sharing in cloud networks. We consider a scenarios where the system model is known but the number of users is very large. In such scenario we analyze the structure of the reward function and implement iterative algorithms which are able to the equilibrium point. Although we consider cloud networks, our resulting analysis is more general and is applicable to similar resources sharing networks.

The contribution of this work is listed below.

- We consider cloud resource sharing games with continuous action spaces where each user tries to maximize its own payoff. In the finite regime (finite number of users), we provide closed-form expressions of three important game-theoretic solution concepts: Nash equilibrium (NE) , finite population mean-field response (F-MFR) and finite population evolutionarily stable strategies (F-ESS). The first surprising result is that the Nash equilibrium is not an evolutionarily stable strategy for finite population. (see Proposition 2). In fact in cloud networks, the number of users as well as users' demands are constantly changing and in this context evolutionary

stable strategy (i.e. it cannot be invaded by small fraction of deviants) is crucial. Based on the work of Possajennikov (2003, [Pos03]) we show that mean-field-taking strategies are evolutionarily stable strategies for a certain range of return index (i.e. α). We provide the explicit optimal pricing as a function of the number of active users. Under the optimal pricing no resource is wasted at the equilibrium. This means that the efficiency loss tends to zero.

- In the finite regime, we then provide an Ishikawa-based iterative algorithm that converges to each of the three aforementioned equilibrium concepts and discuss its convergence time.
- We then consider the infinite regime, because in many cloud networks the number of users is expected to be very high. We provide closed-form expression of the mean-field equilibrium and show that the mean-field equilibrium is an evolutionarily stable strategy, i.e., it cannot be invaded by a small fraction of deviants. We show that the mean-field equilibrium is evolutionary stable for any fraction of deviants strictly less than 100% which makes our result even stronger. We also show that Nash equilibrium is stable in the infinite regime. This is due to the fact that in the limiting case, the influence of one generic player on the total demand is negligible.
- Then, we focus on distributed learning algorithm for infinite regime. We propose a model-based (but still with less information) learning scheme for games with continuous action space and large number of players. Each player will update its learning strategy based on an aggregative term, which is the mean of an increasing function of the action of the other players. Each player will be influenced by the aggregate, and the mean-field behavior is formed from the action of each player. Each player will try to conjecture the aggregative term consisting of the actions of the other players at each time slot, and will respond to the aggregative term locally. This drastically simplifies the dimensionality of the mean-field response system in the asymptotic case. We also discuss the convergence time of the proposed learning algorithm and show that it is faster than the Ishikawa-based algorithm or standard contraction mapping fixed-point algorithms.
- Finally we study of satisfactory game concept. Each cloud user has a capacity demand that needs to be satisfied anytime anywhere (instead of

trying to maximize a payoff). In the finite regime, we provide a closed form expression of the satisfactory solution and develop a distributed algorithm which is able to reach it. Contrary to the aforementioned equilibrium concepts, the extension of the obtained satisfactory solutions to the infinite regime is straightforward.

In chapter 4 we provide a generalized conclusion of the work that has been presented along with implications and applications to future networks.

1.2.3 Publication List

Here we list some of the publications related to this thesis.

- HTAZ13Ja A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Cloud networking mean-field games. *IEEE Journal on Networking*, (submitted for publication), 2013.
- HTAZ13Jb A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. On the convergence of a nash seeking algorithm with stochastic state dependent payoffs. *IEEE Transactions on Automatic Control* (submitted for publication), 2013.
- HTAZ13a A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed power control in femto cells using bayesian density tracking. In *Allerton*, 2013.
- HTAZ13b A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed transmit beamforming with 1-bit feedback for los-miso channels. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th International Workshop on*, june 2013.
- HTAZ12a A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Cloud networking mean field games. In *Cloud Networking (CloudNet), 2012 IEEE 1st International Conference on*, nov 2012.
- HTAZ12b A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed stochastic learning for continuous power control in wireless networks. In *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*, june 2012.

Chapter 2

Model Free Nash Seeking

2.1 Introduction

In this work we consider a fully distributed interactive system, which consists of nodes which can be modeled as a strategic decision making problem. Let us consider a distributed interactive system with N nodes or agents which interact with one another and each has a payoff/utility/reward to maximize. The decision or action of each node has an impact on the payoff of the other nodes, which makes the problem challenging in general. In addition, there is a random variable (scalar, vector or matrix) that could influence the payoff of each node. In distributed interactive systems it might not be possible to have a bird's eye view of the system as it is too complicated or is constantly changing. For practical reason in terms of observation and measurements, we work on discrete time space. We consider interactive systems where each node only has access to a numerical value of its payoff at each time. Let $a_{j,k}$ be the action of node j at time k and the numerical value of the payoff of this node is given by $\tilde{r}_{j,k}$, which is a realization of some random variable $\tilde{r}_{j,k} := r_j(\mathbf{S}_k, \mathbf{a}_k) + \eta_{j,k}$ where $\eta_{j,k}$ represents noise, $r_j : \mathcal{S} \times \mathbb{R}_+^N \rightarrow \mathbb{R}$ is the payoff function of node j , $\mathbf{S}_k \in \mathcal{S} \subseteq \mathbb{C}^{N \times N}$ is the state (which is stochastic), $\mathbf{a}_k = (a_{1,k}, \dots, a_{N,k})$ is the action profile at time k i.e., the vector containing actions of all nodes at time k . Figure 2.1 shows the system model where we have N interacting nodes.

The payoffs are interdependent as the nodes interact with one another. Each of these nodes j has access to the numerical value of their respective payoff $\tilde{r}_{j,k}$ and it needs to implement a scheme to select an action $a_{j,k}$ such that its payoff is maximized. The above scenario can be interpreted as a strategic decision making

problem, i.e. a game. In this work we explore learning in such games which is synonymous with designing distributed iterative algorithms that converge to a solution. One of most known solution concepts in game theory is the so-called Nash equilibrium. For the clarity of the presentation, we restrict ourselves to Nash equilibrium seeking, however, the methodology presented here can be applied to satisfactory solution seeking or correlated equilibrium seeking. Figure 2.1 shows the nodes interacting with one another through a dynamic environment where state \mathbf{S} is stochastic. One of the main assumptions that we can make here is the existence of a local solution.

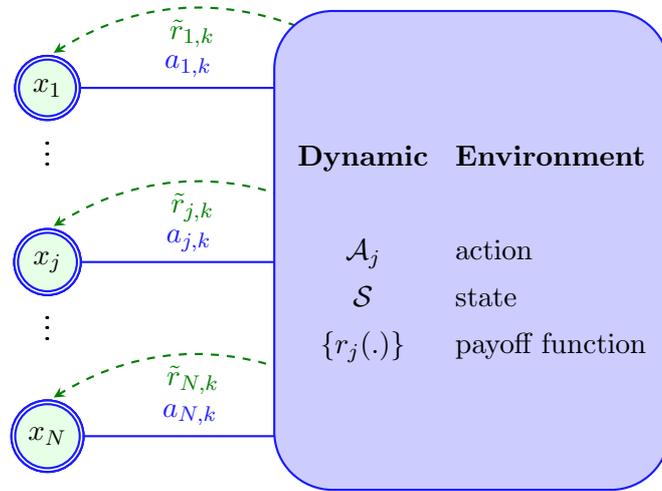


Figure 2.1: Nodes interacting with each other through a dynamic environment

To solve optimization problems there are various approaches, mainly based on gradient ascent or descent method [Sny05], have been developed to achieve a local optimum (or global optimum in some special cases, e.g. concavity of the payoff, etc.) of the distributed optimization problem. The stochastic gradient ascent proposes to feedback the numerical value of gradient of payoff function ∇r_j of node j (which can be noisy) to itself. This supposes in advance that a noisy gradient can be computed or is available at each node. Note that if the numerical value of the gradients of the payoffs (the entire vector) are not known by the nodes at each step, this scheme cannot be used. In [BJ12] projected stochastic gradient based algorithm is presented. A distributed asynchronous stochastic gradient optimization algorithms is presented in [TBA86]. Incremental sub-gradient Methods for Non-differentiable optimization are discussed in [NB01]. A

distributed optimization algorithms for sensor networks is presented in [RN04]. Interested readers are referred to a survey by Bertsekas [Ber10] on incremental gradient, sub-gradient, and proximal methods for convex optimization. In [SS09] the authors present stochastic extremum seeking with applications to mobile sensor networks.

Use of the aforementioned gradient based methods requires the knowledge of:

1. the system state.
2. the actions of others and their states.
3. the mathematical structure (closed form expression) of the payoff function.
4. computational capability of the nodes.

As we can see, it will be difficult for node j to compute the gradient if the expression for the payoff function $r_j(\cdot)$ is unknown and/or if the states and actions of other nodes are not observed as $r_j(\cdot)$ depends on the actions and states of others.

There are several methods for Nash equilibrium seeking where we only have access to the numerical value of the function at each time and not its gradient (e.g. complex functions which cannot be differentiated or unknown functions). A promising technique for solving the above problem is called extremum seeking.

Extremum Seeking

Krstic et.al. in recent years have contributed greatly to the field of *non-model based* extremum seeking. In [FKB12b], the authors propose a Nash seeking algorithm for games with continuous action spaces. They proposed a fully distributed learning algorithm which requires only a measurement of the numerical value of the payoff. Their scheme is based on sinus perturbation (i.e. deterministic perturbation instead of stochastic perturbation) of the payoff function in continuous time. However, discrete time learning scheme with sinus perturbations is not examined in [FKB12b]. In [SS10] extremum seeking algorithm with sinusoidal perturbations for *non-model based* systems has been extended and modified to the case of i.i.d. noisy measurements and vanishing sinus perturbation, almost sure convergence to equilibrium is proved. *Sinus perturbation*

based extremum seeking for *state independent noisy measurement* is presented in [SJS12].

In [vdWHN12] an extremum-seeking approach for periodic steady-state response optimization is presented. A unifying framework for analysis and design of extremum seeking controllers is presented in [NTM⁺12]. In [NNTM12] a shubert algorithm-based global extremum seeking scheme is presented. A multidimensional global extremum seeking via the DIRECT method is presented in [KMNT12]. In [LK12] a Newton-based stochastic extremum seeking approach is presented. A dynamic bandwidth allocation framework for wireless networks using a Shahshahani gradient based extremum seeking control is presented in [PQ12a]. A simplex guided extremum seeking control for real-time optimization is presented in [ZG12]. A Shahshahani Gradient based extremum seeking scheme is presented in [PQ12b]. Extremum seeking under input constraint for systems with a time-varying extremum is presented in [YH13]. In [AJG13] an on-line optimization of cone crushers using Extremum-Seeking Control is considered. In [NMM13] a framework for extremum seeking control of systems with parameter uncertainties is considered. A time-varying extremum-seeking control approach is presented in [GDD13]. Examples of distance-based synchronization: An extremum seeking approach is presented in [DSJE13]. In [PQ13] a distributed extremum seeking for real-time resource allocation is presented. An extremum-seeking control of ABS braking in road vehicles with lateral force improvement is presented in [DGA14].

In [LK10] Krstic et al. have recently extended Nash seeking scheme to stochastic non-sinusoidal perturbations and convergence in probability is proved. Interested readers can also refer to the references therein [LK10] for other perturbation method. In [LK10] stochastic averaging theory is developed for extremum seeking problems and several conditions on the payoff functions like global Lipschitz condition has been relaxed. Notions of weakly stable, weakly asymptotically stable and weakly exponentially stable are defined for extremum seeking. In [LK11], building on the theoretical foundations in [LK10], Krstic et al. present stochastic perturbations based Nash equilibrium seeking for games with non linear payoffs and they analyze necessary convergence conditions. Although these stochastic perturbation techniques do estimate the gradient but they introduce a level of uncertainty because of the stochastic perturbations. To avoid this, sinus perturbation can be used instead of stochastic perturbation in

certain applications. Deterministic (sinus) perturbation method is particularly helpful when one node is trying to track the actions of the other nodes in many engineering application, one such application is presented here.

In this work we extend the work in [FKB12b] to the case of *stochastic state dependent* payoff functions, and use deterministic perturbations for Nash seeking. One can see easily the difference between this work and the existing works [SS10, SJS12]. In these works, the noise η_j associated with the measurement is additive i.i.d. which does not necessarily hold in practice especially in engineering application where the noise is in general time correlated. In our case, we consider a stochastic state dependent payoff function and our problem can be written in Robbins-Monro form with a Markovian (correlated) noise given by $\eta_j = r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}}[r_j(\mathbf{S}, \mathbf{a})]$ (this will become clearer in the next sections), i.e. the associated noise is stochastic state-and-action dependent which is different from the case of additive i.i.d. noise assumed in [SS10, SJS12].

In this work, we propose a *discrete time* learning algorithm, using sinus perturbation, for *continuous* action games where each node has only a numerical realization of own-payoff at each time. We therefore extend the classical Nash Seeking with sinus perturbation method [FKB12b] to the case of discrete time and stochastic state-dependent payoff functions. We prove that our algorithm converges locally in the close neighborhood of a state independent Nash equilibrium in Theorem 1 for vanishing step size and provide an error bound in Theorem 2 for fixed step size. Note that since the payoff function may not necessarily be concave, finding a global optimum in reasonable time can be difficult in general even in deterministic case (fixed state) and known closed-form expression of payoff. Also note that, our convergence is in the sense of Borkar [Bor08] and will be explained later on in the text. We also show the convergence time for the sinus framework in Corollary 1.

The proof of the theorems are given in Appendix A.

The remainder of this work is organized as follows. Section 2.2 provides the proposed distributed stochastic learning algorithm. The performance analysis of the proposed algorithm (convergence to ODE, error bounds) is presented in Section 2.3. A numerical example with convergence plots is provided in Section 2.4. Appendix A contains the proofs.

We summarize some of the notations in Table 2.1.

Table 2.1: Summary of Notations

Symbol	Meaning
\mathcal{N}	set of nodes
\mathcal{A}_j	set of choices of node j ,
\mathcal{S}	state space
r_j	payoff of node j
$a_{j,k}$	decision of j at time k
$\mathbf{a}_{-j,k}$	$(a_{j',k})_{j' \neq j}$
\mathbf{S}_k	state at time k
\mathbb{E}	expectation operator
∇	gradient operator

2.2 Problem Formulation and Proposed Algorithm

Let there be N distributed nodes, each with a payoff function represented by $r_j(\mathbf{S}_k, a_{j,k}, \mathbf{a}_{-j,k})$ at time k which is used to formulate the following robust problems:

$$\sup_{a_j \geq 0} \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, a_j, \mathbf{a}_{-j}) \quad j \in \mathcal{N} \quad (2.1)$$

where $\mathcal{N} := \{1, \dots, N\}$ is the set of nodes, $\mathcal{A}_j \subseteq \mathbb{R}$ is the action space of node j , \mathcal{S} is the state space of the whole system, where $\mathcal{S} \subseteq \mathbb{C}^{N \times N}$ and $r_j : \mathcal{S} \times \prod_{j' \in \mathcal{N}} \mathcal{A}_{j'} \rightarrow \mathbb{R}$ is a smooth function. A solution to the problem (2.1) is called *state-independent* equilibrium in a game-theoretic setting. Games with state uncertainties are often called *robust games*. Since the state is stochastic, we get a robust game. It should be mentioned here for clarity that the decisions are taken in a decentralized fashion by each node.

A solution to 2.1 is a state-independent equilibrium in the sense that no node has an incentive to change its action when the other nodes keep their choice and the equilibrium strategy does not depend on the state. It may depend on the entire distribution of states.

Definition 1 (Nash Equilibrium (state-independent)). *The action profile $\mathbf{a}^* = (a_j^*, \mathbf{a}_{-j}^*) \in \prod_{j'} \mathcal{A}_{j'}$ is a (state-independent) Nash equilibrium point if*

$$\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, a_j^*, \mathbf{a}_{-j}^*) \geq \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, a_j', \mathbf{a}_{-j}^*), \quad (2.2)$$

where $\forall a'_j \in \mathcal{A}_j$, $a'_j \neq a_j^*$, $\mathbb{E}_{\mathbf{S}}$ denotes the mathematical expectation over the state.

It is well-known that equilibria can be different than global optima, the gap between the worse equilibrium and the global optimum is captured by the so-called *price of anarchy*. Thus, solution obtained by our method can be suboptimal with respect to maximizing the sum of all the payoffs. We study the local stability of the stochastic algorithm. Here we denote $\mathbf{a} := (a_j, \mathbf{a}_{-j})$, assuming that node j has access to it's realized payoff at each time k but the closed-form expression of $r_j(\mathbf{S}_k, a_{j,k}, \mathbf{a}_{-j,k})$ is unknown to node j .

Definition 2 (Nash Equilibrium (state-dependent)). We define a state-dependent strategy \check{a}_j of node j as a mapping from \mathcal{S} to the action space $\check{\mathcal{A}}_j$. The set of state-dependent strategy is $\mathcal{PG}_j : \{\check{a}_j : \mathcal{S} \rightarrow \check{\mathcal{A}}_j, \mathbf{S} \mapsto \check{a}_j(\mathbf{S}) \in \check{\mathcal{A}}_j\}$.

$$\check{\mathbf{a}}^* = (\check{a}_j^*, \check{\mathbf{a}}_{-j}^*) \in \prod_i \mathcal{PG}_i$$

is a (state-dependent) Nash equilibrium point if

$$\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \check{a}_j^*(\mathbf{S}), \check{\mathbf{a}}_{-j}^*(\mathbf{S})) \geq \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \check{a}'_j(\mathbf{S}), \check{\mathbf{a}}_{-j}^*(\mathbf{S})), \quad (2.3)$$

$$\forall \check{a}'_j \in \mathcal{PG}_j.$$

Here we will focus on the analysis of the so-called *expected robust game* i.e., $(\mathcal{N}, \mathcal{A}_j, \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \cdot))$. A (state-independent) Nash equilibrium point [Jr.50] of the above robust game is a strategy profile such that no node can improve its payoff by unilateral deviation, see Definition 1 and Definition 2.

Since the current state is not observed by the nodes, it will be difficult to implement state-dependent strategy. Our goal is to design a learning algorithm for a state-independent equilibrium given in Definition 1. In what follows we assume that we are in a setting where the above problem has at least one isolated state-independent equilibrium solution. More details on existence of equilibria can be found in Theorem 3 in [BTZ93].

2.2.1 Distributed learning algorithm

Suppose that each node j is able to observe a numerical value $\tilde{r}_{j,k}$ of the function $r_j(\mathbf{S}_k, \mathbf{a}_k)$ at time k , where $\mathbf{a}_k = (a_{j,k}, \mathbf{a}_{-j,k})$ is the action of nodes at time k . $\hat{a}_{j,k}$ is an intermediary variable. The real numbers b_j Ω_j ϕ_j respectively represent

the amplitude, frequency and phase of the sinus perturbation signal given by $b_j \sin(\Omega_j t_k + \phi_j)$, The learning algorithm is presented in Algorithm 2 and is explained below. At each time instant k , each node updates its action $a_{j,k}$, by adding the sinus perturbation i.e. $b_j \sin(\Omega_j t_k + \phi_j)$ to the intermediary variable $\hat{a}_{j,k}$ using equation (2.4), and makes the action using $a_{j,k}$. Then, each node gets a realization of the payoff $\tilde{r}_{j,k+1}$ from the dynamic environment at time $k + 1$ which is used to compute $\hat{a}_{j,k+1}$ using equation (2.5). The action $a_{j,k+1}$ is then updated using equation (2.4). This procedure is repeated for the window T .

The algorithm is in discrete time and is given by

$$a_{j,k} = \hat{a}_{j,k} + b_j \sin(\Omega_j t_k + \phi_j) \quad (2.4)$$

$$\hat{a}_{j,k+1} = \hat{a}_{j,k} + \lambda_k z_j b_j \sin(\Omega_j t_k + \phi_j) \tilde{r}_{j,k+1} \quad (2.5)$$

where $t_k := \sum_{\hat{k}=1}^k \lambda_{\hat{k}}$, $\Omega_j \neq \Omega_{j'}, \Omega_{j'} + \Omega_j \neq \Omega_{j''} \forall j, j', j''$.

For almost sure (a.s.) convergence, it is usual to consider vanishing step-size or learning rate such as $\lambda_k = \frac{1}{k+1}$. However, constant learning rate $\lambda_k = \lambda$ could be more appropriate in some regime. The parameter ϕ_j belongs to $[0, 2\pi] \forall j$, $k \in \mathbb{Z}_+$

- 1: Each node j , initialize $\hat{a}_{j,0}$
- 2: Repeat
- 3: Calculate action $a_{j,k}$ according to Equation (2.4)
- 4: Perform action $a_{j,k}$
- 5: Observe $\tilde{r}_{j,k}$
- 6: Update $\hat{a}_{j,k+1}$ using Equation (2.5)
- 7: until horizon T

Algorithm 2: Distributed learning algorithm

Remark 1 (Learning Scheme in Discrete Time). *As we will prove in subsection 2.3, the difference equation (2.5) can be seen as a discretized version of the learning scheme presented in [FKB12b]. But it is for games with state-dependent payoff functions i.e., robust games.*

It should be mentioned here for clarity that the action $a_{j,k}$ of each node j is scalar.

2.2.2 Interpretation of the proposed algorithm

In some sense our algorithm is trying to estimate the gradient of the function $r_j(\cdot)$, note that we don't have access to the function but just its numerical value.

The following equation clearly illustrated the significance of each variable and constant in the algorithm.

$$\underbrace{\hat{a}_{j,k+1}}_{\text{New Value}} = \underbrace{\hat{a}_{j,k}}_{\text{Old Value}} + \underbrace{\lambda_k}_{\text{Learning Rate}} \underbrace{z_j}_{\text{Growth Rate}} \underbrace{b_j}_{\text{Perturbation Amplitude}} \sin\left(\underbrace{\Omega_j}_{\text{Perturbation Frequency}} t_k + \underbrace{\phi_j}_{\text{Phase}}\right) \underbrace{\tilde{r}_{j,k+1}}_{\text{New Reward}} \quad (2.6)$$

The learning rate λ_k can be constant or variable depending on the requirements of the algorithm and system limitations. Perturbation amplitude $b_j > 0$ is a small number. $z_j > 0$ is also a small value which can be varied for fine tuning. Rewriting the above equation we get

$$\frac{\hat{a}_{j,k+1} - \hat{a}_{j,k}}{\lambda_k} = z_j b_j \sin(\Omega_j t_k + \phi_j) \tilde{r}_{j,k+1} \quad (2.7)$$

For vanishing step size as $k \rightarrow \infty$ $\lambda_k \rightarrow 0$ and the trajectory of the above algorithm coincides with the trajectory of an ODE.

For a survey on the above sinus perturbation method see [FKB12b].

2.3 Main results

In this section we present the convergence results as introduced in the contribution section.

We introduce the following assumptions that will be used step by step¹.

Assumption 1 (A1: Vanishing learning rate). $\lambda_k > 0$, $\sum_k \lambda_k = \infty$, $\sum_k |\lambda_k|^2 < \infty$. There exists $C_0 > 0$ such that $\mathbb{P}(\sup_k \|a_k\| < C_0) = 1$. λ_k represents the step size of the algorithm, The reason for first assumption: $\sum_k \lambda_k = \infty$, is that we need to traverse over all discrete time. The condition $\sum_k |\lambda_k|^2 < \infty$ ensures bound for the cumulative noise error. This last assumption is for a local stability analysis.

Assumption 2 (A2: Constant learning rate). $\lambda_t = \lambda > 0$, $\sup_t [\mathbb{E}\|\mathbf{a}_t\|^2]^{\frac{1}{2}} < +\infty$ and $\|\mathbf{a}_t\|^2$ is uniformly integrable.

For Lipschitz functions the following two conditions must hold for the existence of a Nash equilibrium.

¹ A1 and A2 are not used simultaneously.

Assumption 3 (A3: Existence of a local best response). $\mathbb{E}_{\mathbf{S}} \frac{\partial r_j(\mathbf{S}, \mathbf{a}^*)}{\partial a_j} = 0$, $\mathbb{E}_{\mathbf{S}} \frac{\partial^2 r_j(\mathbf{S}, \mathbf{a}^*)}{\partial a_j^2} < 0$. These two conditions tell us that a_j^* is a local maximizer of $a_j \rightarrow \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, a_j, \mathbf{a}_{-j}^*)$ where $\mathbf{a}_{-j}^* = (a_1^*, \dots, a_{j-1}^*, a_{j+1}^*, \dots, a_N^*)$.

Assumption 4 (A4: Diagonal Dominance). the expected payoff has a Hessian that is diagonally dominant at \mathbf{a}^* , i.e., $\left| \mathbb{E}_{\mathbf{S}} \left(\frac{\partial^2 r_j(\mathbf{S}, \mathbf{a}^*)}{\partial a_j^2} \right) \right| - \sum_{j' \neq j} \left| \mathbb{E}_{\mathbf{S}} \left(\frac{\partial^2 r_j(\mathbf{S}, \mathbf{a}^*)}{\partial a_j \partial a_{j'}} \right) \right| > 0$. Note that A4 implies that the Hessian of the expected payoff is invertible at \mathbf{a}^* . This assumption is weaker compared to the classical extremum seeking algorithm because the Hessian of $r_j(\mathbf{S}, \mathbf{a}^*)$ does not need to be invertible for each \mathbf{S} .

We assume $\mathbf{S} \mapsto r_j(\mathbf{S}, \mathbf{a})$ is integrable with respect to \mathbf{S} so that the expectation $\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is finite.

Assumption 5 (A5). We assume $\mathbf{S} \mapsto r_j(\mathbf{S}, \mathbf{a})$ is integrable with respect to \mathbf{S} so that the expectation $\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is finite.

Assumption 6 (A6). For any given state \mathbf{S} , we assume that $r_j(\mathbf{S}, \mathbf{a})$ is a smooth function with respect to \mathbf{a} .

Convergence to ODE

First we need to show that our proposed algorithm converges to the respective ordinary differential equation (ODE) almost surely. We will use a dynamical system viewpoint and a stochastic approximation method to analyze our learning algorithm. The idea consists of finding the asymptotic pseudo-trajectory of the algorithm via ODE. To do so, we use the framework initiated by Robbins-Monro [RM51] or [KW52]. See [Ben99, Bor08] for recent developments. The works in [Ben99, Bor08] allows us to find the limiting trajectory of the learning algorithm.

The scheme can be written as $\hat{a}_{j,k+1} = \hat{a}_{j,k} + \lambda_k z_j b_j \sin(\Omega_j t_k + \phi_j) \tilde{r}_{j,k+1}$. Now we rewrite the above equation in Robbins-Monro [RM51] form as:

$$\hat{a}_{j,k+1} = \hat{a}_{j,k} + \lambda_k [f_j(k, \hat{\mathbf{a}}_k) + M_{k+1}],$$

where

$$\begin{aligned} f_j(k, \mathbf{a}_k) &:= z_j b_j \sin(\Omega_j t_k + \phi_j) \mathbb{E}_{\mathbf{S}} \hat{r}_j(\mathbf{S}, \mathbf{a}_k) \\ \hat{r}_j(\mathbf{S}, \hat{\mathbf{a}}_k) &:= r_j(\mathbf{S}, (\hat{a}_{j,k} + b_j \sin(\Omega_j t_k + \phi_j))_{j \in \mathcal{N}}) \\ M_{k+1} &:= z_j b_j \sin(\Omega_j t_k + \phi_j) [\tilde{r}_{j,k+1} - \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}_k)] \\ \mathbf{f}(\cdot) &:= (f_j(\cdot))_{j \in \mathcal{N}} \end{aligned}$$

Since the payoff r_j is Lebesgue integrable with respect to \mathbf{S} , the expectation of the payoff function $\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is finite. M_{k+1} is a Martingale difference sequence.

The following equations (2.8) (2.9) represent the non-autonomous system of ODEs for Lipschitz functions.

$$\frac{d}{dt} \hat{a}_{j,t} = z_j b_j \sin(\Omega_j t + \phi_j) \mathbb{E}_{\mathbf{S}} [r_j(\mathbf{S}, \mathbf{a}_t)] \quad (2.8)$$

$$a_{j,t} = \hat{a}_{j,t} + b_j \sin(\Omega_j t + \phi_j) \quad (2.9)$$

Theorem 1 (Variable Learning Rate). *Under Assumptions A1, A5 and A6 the interpolated trajectory of our learning algorithm converges almost surely to the trajectory of a non-autonomous system given by equation (2.8) and (2.9) in the following sense,*

$$\lim_{t_k \rightarrow \infty} \sup_{t \in [t_k, t_k + T]} \|\bar{\mathbf{a}}_t - \mathbf{a}_t^{t_k}\| = 0 \text{ a.s.}$$

Almost sure (a.s.) convergence in our case simply means that the interpolated trajectory of our proposed algorithm converges to the solution of the ODE almost surely i.e. convergence with probability 1. Please refer to [Kus03] for more details on almost sure convergence.

The gap between the interpolated version $\bar{\mathbf{a}}_t$ of algorithm and the trajectory of the ODE given by \mathbf{a}^{t_k} is bounded as

$$\sup_{t \in [t_k, t_k + T]} \|\bar{\mathbf{a}}_t - \mathbf{a}_t^{t_k}\| \leq K_{T,k} e^{LT} + C_T \sup_{\acute{k} \geq 0} \lambda_{k+\acute{k}} \quad (2.10)$$

which vanishes, where $\bar{\mathbf{a}}_t := \mathbf{a}_k + (\mathbf{a}_{k+1} - \mathbf{a}_k) \frac{(t-t_k)}{(t_{k+1}-t_k)}$ is the interpolated version of the algorithm and $\mathbf{a}_t^{t_k}$ is the solution of the ODE at time t starting from $t_k := \sum_{\acute{k}=1}^k \lambda_{\acute{k}}$, where L is the Lipschitz constant for the ODE and T is the time window. $K_{T,k}$ and C_T are specified below.

In order to calculate the bound in equation (2.10) we need to define a few

terms which are helpful in obtaining a compact form of the bound.

$$K_{T,k} := C_T L \sum_{\acute{k} \geq 0} \lambda_{k+\acute{k}}^2 + \sup_{\acute{k} \geq 0} \|\delta_{k,k+\acute{k}}\| \quad (2.11)$$

$$\delta_{k,k+\acute{k}} := \xi_{k+\acute{k}} - \xi_k \quad (2.12)$$

$$\xi_k := \sum_{\acute{k}=0}^{k-1} \lambda_{k-\acute{k}} M_{k-\acute{k}+1} \quad (2.13)$$

$$C_T := \|\mathbf{r}_0\| + L(C_0 + \|\mathbf{r}_0\|T)e^{LT} < \infty \quad (2.14)$$

$$L := \max_{j \in \mathcal{N}} \mathbb{E}_{\mathbf{S}}[L_j, \mathbf{S}] \quad (2.15)$$

$$\mathbf{r}_k := [r_{1,k}, \dots, r_{N,k}] \quad (2.16)$$

To prove that the learning algorithm (the learning algorithm (discrete ODE)) converges to the ODE we need to verify conditions from Borkar [Bor08] Chapter 2 Lemma 1 for non-autonomous case.

This is an important result as it gives us an approximation on the error between our algorithm and the corresponding ODE. The proof of theorem 1 is available in the Appendix A.1.

Theorem 2 (Fixed Learning Rate). *Under Assumption A2, A5 and A6 the interpolated trajectory of the learning algorithm converges in distribution when $\lambda \rightarrow 0$, to the solution of a non-autonomous system given by equation (2.8) and (2.9). Moreover the error gap is in order of λ .*

The advantage of Theorem 2 compared to Theorem 1 is the convergence time. The number of iterations required to reach a fixed time T is less with constant learning rate than the vanishing learning rate. However, the convergence notion under constant step size is weaker (it is in distribution) compared to the almost surely convergence with vanishing learning rate. So there is a sort of tradeoff between almost sure convergence and convergence time. Please refer to [Kus03] for more information on convergence in distribution.

Let $\Delta_t := \|\mathbf{a}_t - \mathbf{a}^*\|$ be the gap between the trajectory of the ODE at time t and the isolated equilibrium.

Theorem 3 (Local Exponential Stability). *Assume A3-A6. Then, there exist $\acute{M}, \acute{T} > 0$ and $\bar{\epsilon}, \bar{b}_j$ such that, for all $\epsilon \in (0, \bar{\epsilon})$ and $b_j \in (0, \bar{b}_j)$, if the initial gap is $\Delta_0 := \|\mathbf{a}^* - \mathbf{a}_0\|$ (which is small) then for all time t ,*

$$\Delta_t \leq y_{1,t} \quad (2.17)$$

where

$$y_{1,t} := \acute{M}e^{-\acute{T}t}\Delta_0 + O(\epsilon + \max_j b_j^3) \quad (2.18)$$

The proof of this theorem is presented in the appendix [A.3](#).

According to [Theorem 3](#) we have convergence to within a small range of the equilibrium point i.e. we do not converge to the equilibrium point.

From the above equation it is clear that as time goes to infinity the first term in $y_{1,t}$ bound vanishes exponentially and the error is bounded by the amplitude of the sinus perturbation i.e. $O(\epsilon + \max_j b_j^3)$. This means that the solution of ODE converges locally exponentially to the neighborhood of the state-independent equilibrium action \mathbf{a}^* provided the initial solution is relatively close. The initial error Δ_0 influences the choice of $\bar{\epsilon}$ in [Theorem 3](#).

Definition 3 (ϵ -Nash equilibrium payoff point). *An ϵ -Nash equilibrium payoff point in state-independent strategy is a strategy profile such that no node can improve its payoff more than ϵ by unilateral deviation.*

Definition 4 (ϵ -close strategy Nash equilibrium point). *An ϵ -close strategy Nash equilibrium point in a state-independent strategy is a strategy profile such that the Euclidean distance to a Nash equilibrium is less than ϵ .*

A ϵ -close strategy Nash equilibrium point is an approximate Nash point with a precision at most ϵ .

It is not difficult to see that for Lipschitz continuous payoff functions, an ϵ -close Nash equilibrium is an $L\epsilon$ -Nash equilibrium point where L is the Lipschitz constant.

Next corollary shows that one can get an ϵ -close strategy Nash equilibrium in finite time.

Corollary 1 (Convergence Time). *Assume [A3-A4](#) and [Remark 6,7](#) holds. Then, the ODE reaches a $(2\epsilon + \max_j b_j^3)$ -close to a Nash equilibrium in at most T time units where $T = \frac{1}{\acute{T}} \log(\frac{\Delta_0 \acute{M}}{\epsilon})$*

Sketch of Proof for [Corollary 1](#). The proof follows from the inequality [\(2.17\)](#) in [Theorem 3](#).

Corollary 2 (Convergence). *Under [Assumption A1, A3, and A4](#), the following inequality holds almost surely: $\|\bar{\mathbf{a}}_t - \mathbf{a}^*\| \leq y_{1,t} + y_{2,t}$*

where

$$y_{2,t} := C_T(\lambda_{k+k} + L \sum_{\hat{k} \geq 0} \lambda_{k+\hat{k}}^2) + \sup_{\hat{k} \geq 0} \|\delta_{k,k+\hat{k}}\| \quad (2.19)$$

and $t \in [t_k, t_k + T]$

Proof of Corollary 2. The proof uses the triangle inequality $\|\bar{\mathbf{a}}_t - \mathbf{a}^*\| \leq \|\bar{\mathbf{a}}_t - \mathbf{a}_t\| + \|\mathbf{a}_t - \mathbf{a}^*\|$. By Theorem 1, we get $\|\bar{\mathbf{a}}_t - \mathbf{a}_t\| \leq y_{2,t}$ and by Theorem 3, we have $\|\mathbf{a}_t - \mathbf{a}^*\| \leq y_{1,t}$. Combining together, we arrive at the announced result.

This completes the proof. \square

The constants in equation (2.18) and (2.19) depends on the number of players and the dimension of the action space.

2.4 Numerical Example A: Wireless Interference

Even though the distributed optimization problem, considered here, and the developed approach are general and can be used in many application domains. As an application of the above framework, we will consider the problem of power control in wireless networks in order to better illustrate our contribution. Consider an interference channel composed of N transmit receiver pairs as shown in Figure 2.2. Each transmitter communicates with its corresponding receiver and incurs an interference on the other receivers. Each receiver feeds back a numerical value of the payoff $\gamma_j(\mathbf{H}, \mathbf{p})$ to its corresponding transmitter.

The problem is composed of transmitter-receiver pairs; all of them use the same frequency and thus generate interference onto each other. Each transmitter-receiver pair has therefore its own payoff/reward/utility function that depends necessarily on the interference exerted by the other pairs/nodes. Since the wireless channel is time varying as well as the interference, the objective is necessarily to optimize in the long-run (e.g. average) the payoff functions of all the nodes. The payoff function of node j at time k is denoted by $r_j(\mathbf{H}_k, \mathbf{p}_k)$ where $\mathbf{H}_k := [h_k(i, j)]$ represents an $N \times N$ matrix containing channel coefficients at time k , $h_k(i, j)$ represents the channel coefficient between transmitter i and receiver j (where $(i, j) \in \mathcal{N}^2$) and \mathbf{p}_k represents the vector containing transmit powers of N transmit-receive nodes. The most common technique used to obtain a local maximum of the nodes' payoff functions is the gradient based descent or ascent method.

Table 2.2: Equivalent Notations for Wireless

Remark 2.		
General	Application	Description
$\tilde{r}_{j,k}$	$\gamma_{j,k}$	utility/payoff of transmitter j at time k
$a_{j,k}$	$p_{j,k}$	action/power of transmitter j at time k
$s_{jj',k}$	$g_{jj',k}$	state/channel gain between transmitter j and receiver j' at time k

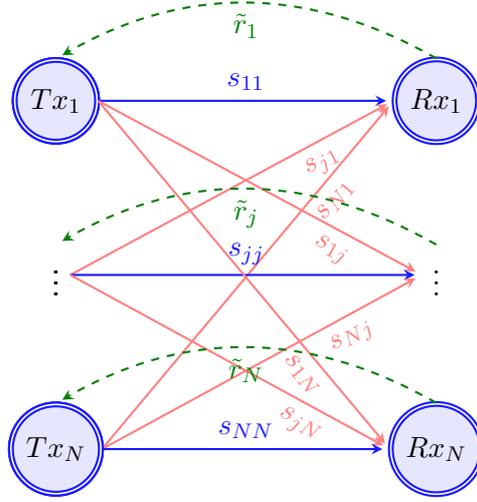


Figure 2.2: Interference Channel Model

In Section 2.3, we proved that our proposed algorithm converges to \mathbf{p}^* for any type of payoff functions which satisfies the assumptions in Section 2.3. In order to show numerically that our algorithm converges to \mathbf{p}^* , we run our algorithm for a simple payoff function. In parallel, we obtain analytically the Nash equilibrium \mathbf{p}^* and compare the convergence point of our algorithm to \mathbf{p}^* . We therefore choose a simple payoff function for which \mathbf{p}^* can be obtained analytically.

The payoff function of node j at time k has then the following form:

$$\gamma_j(\mathbf{H}_k, \mathbf{p}_k) = \underbrace{\omega}_{\text{bandwidth}} \underbrace{\log\left(1 + \frac{\overbrace{SINR_j}^{p_{j,k}g_{jj,k}}}{\sigma^2 + \sum_{j' \neq j} p_{j',k}g_{j'j,k}}\right)}_{\text{Rate}} - \underbrace{\kappa p_{j,k}}_{\text{constraint on powers}} \quad (2.20)$$

where ω represents the bandwidth available for transmission. The above payoff function $\gamma_j(\mathbf{H}_k, \mathbf{p}_k)$ consists of \log of $(1 + SINR_j)$ of user j and the unit cost of

transmission is κ . It is assumed that a user doesn't know the structure function $\gamma_j(\cdot)$ or the law of the channel state. For the above payoff function to ensure the assumption A3-A4 and Remark 6,7 we need to satisfy the condition $\mathbb{E}|h_{jj}|^2 \geq \mathbb{E} \sum_{j' \neq j} |h_{j'j}|^2$. Please see appendix A for more details.

The problem here is to maximize the payoff function $\gamma_j(\mathbf{H}, \mathbf{p})$ which is stated as follows: find \mathbf{p}^* such that for each user $j \in \mathcal{N}$, satisfies $p_j^* \in \arg \max_{p_j \geq 0} \mathbb{E} \gamma_j(\mathbf{H}, p_1^*, \dots, p_{j-1}^*, p_j, p_{j+1}^*, \dots, p_N^*)$. Note that when $g_{jj} = 0$ then the payoff of user j is negative and the minimum power $p_j^* = 0$ is a solution to the above problem. For the remaining, we assume that $|h_{jj}|^2 = g_{jj} > 0$.

The channel $h_{jj'}$ is time varying and is generated using an independent and identically distributed complex Gaussian channel model with variance $\sigma_{jj'}^2$ such that $\sigma_{jj} = 1$ $\sigma_{jj'} = 0.1$, $\forall j' \neq j$. The thermal noise is assumed to be a zero mean Gaussian with variance σ^2 such that $\sigma^2 = 1$.

We consider the following simulation settings with $N = 2$ for the above wireless model: $k_1 = 0.9, k_2 = 0.9, \phi_1 = 0, \phi_2 = 0, \Omega_1 = 0.9, \Omega_2 = 1, b_1 = 0.9, b_2 = 0.9$. The numerical setting could be tuned in order to make the convergence slower or faster with some other tradeoff. Due to space limitations further discussion on how to select these parameters has been omitted. $p_{1,0}$ and $p_{2,0}$ represent the starting points of the algorithm which are initialized as $p_{1,0} = p_1^* + 10$ and $p_{2,0} = p_2^* + 10$. $\kappa = 2$ is the penalty for interference, $\omega = 10$ is the bandwidth and the variance of noise is normalized. Figure 2.3 represents the average transmit power trajectories of the algorithm for two nodes. The dotted line represents \mathbf{p}^* . As can be seen from the plots in Fig 2.3 that the system converges to \mathbf{p}^* where $p_j^* = 3.9604$, $j \in \{1, 2\}$. Fig 2.4 shows that the reward also approaches the optimal value.

The example we discussed is only one of the possible types of applications where our proposed algorithm can be implemented.

Consider for example the following payoffs: $q_1(\cdot) = \text{goodput}(\cdot)$ and $q_2(\cdot) = \mathbb{P}(\text{goodput}(\cdot) < \eta)$ where η is a small value and $\mathbb{P}(\cdot)$ stands for probability. Goodput represents the ratio of correctly received information bits vs the number of transmitter bits. In wireless communications the channel is constantly changing due to various physical phenomenon and interference from other sources and changes in the environment. It is hard to have a closed form expression for $q_1(\cdot)$ due to complexity of the transmitter, receiver and unknown parameters. In practice, at each time k , the receiver has therefore a numerical value of $\text{goodput}(\cdot)$ but

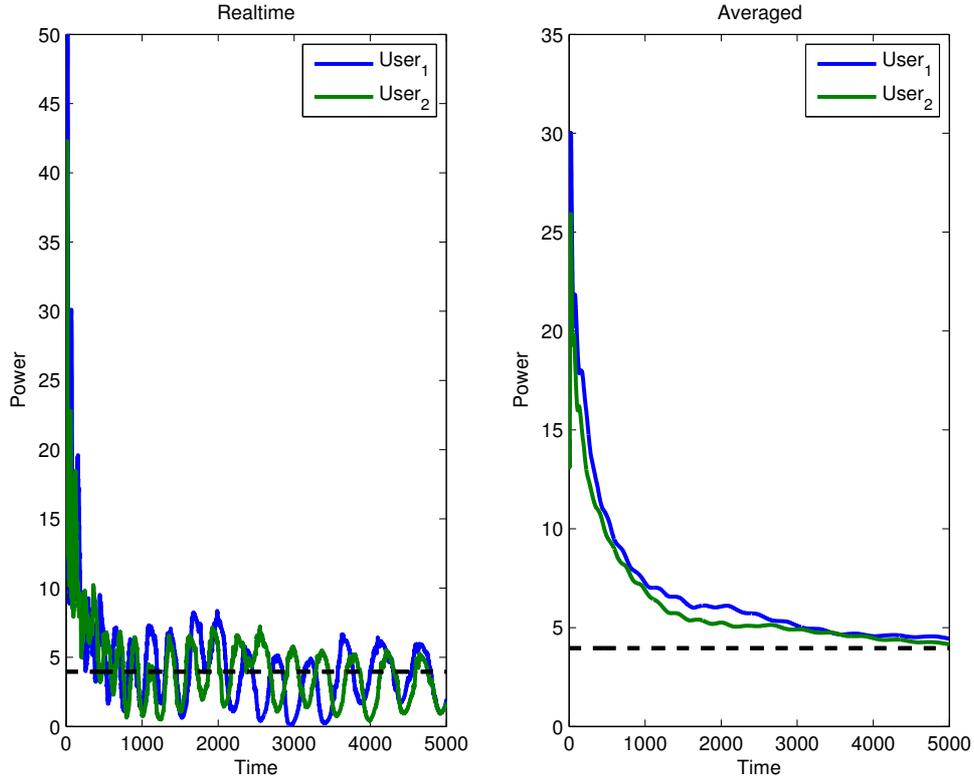


Figure 2.3: Power evolution (discrete time)

no closed form expression for rate/goodput is available especially for advanced coding scheme (e.g. turbo code, etc.). $q_2(\cdot)$ represents an outage probability for which also depends on the goodput, the gradient for $q_2(\cdot)$ is notoriously hard to compute without channel and interference statistics knowledge (probability distribution function) and closed form expression of $goodput(\cdot)$. Our scheme can be particularly helpful in such scenarios.

The price/design parameter κ inside the reward function can be tuned such that the solution of the distributed robust extremum coincides with a global optimizer of the system designer. The κ can be same for all nodes or each node can have its own κ_j . Let \mathbf{a}_j^* represent the optimal action or set of actions to be performed by each node to maximize their respective utilities. It is possible to set κ such that the following equation is satisfied. $\mathbf{a}(\kappa) = \mathbf{a}_j^*$. κ could represent a scalar or a vector depending on the system size and the application. To be able to effectively make $\mathbf{a}(\kappa)$ equal to \mathbf{a}^* we need to have enough degrees of freedom in the system. However this type of tuning is not true in general.

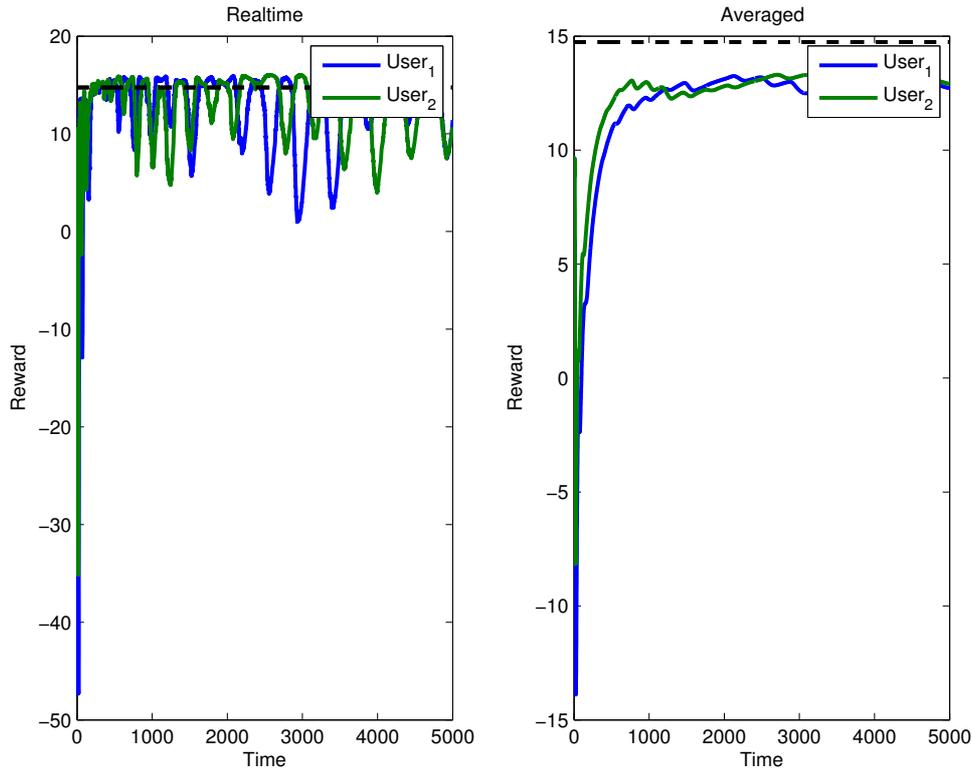


Figure 2.4: Payoff evolution (discrete time)

The plots in Fig 2.5 compare the performance of the power control interference channel when using the the sinus perturbation scenario with the stochastic perturbation scenario.

2.5 Numerical Example B: Distributed Beamforming

We consider a system model with M transmitter–receiver pairs as shown in Fig 2.6 (where $M = 2$ for simplicity), where each access point AP_j is equipped with N_t transmitting antennas and each user U_j has one receiving antenna. Furthermore, we assume that the transmitters and receivers are not mobile. Let $\mathbf{a}_{ij} \in \mathbb{C}^{1 \times N_t}$ represent the complex channel between transmitter AP_j and user $U_i \forall i, j \in \mathcal{M}^2$, where $\mathcal{M} := \{1, \dots, M\}$.

The channel between all transmitter and receivers is assumed to be line of sight (LoS) and frequency flat. From [KSL07] we know that Vandermonde

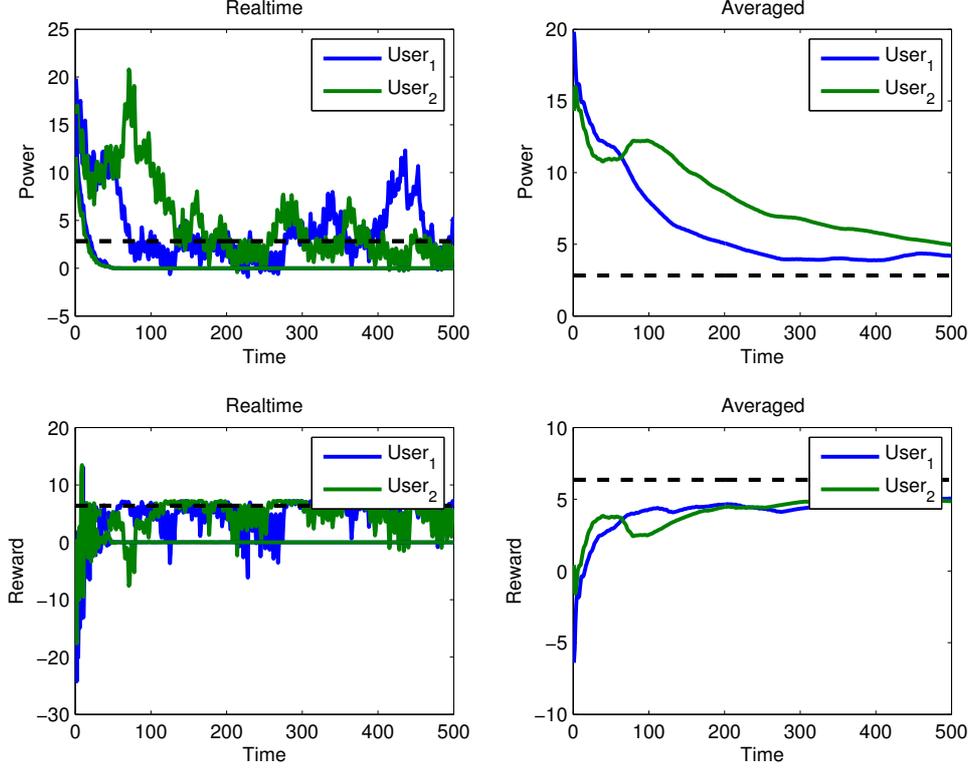


Figure 2.5: Stochastic Perturbation method

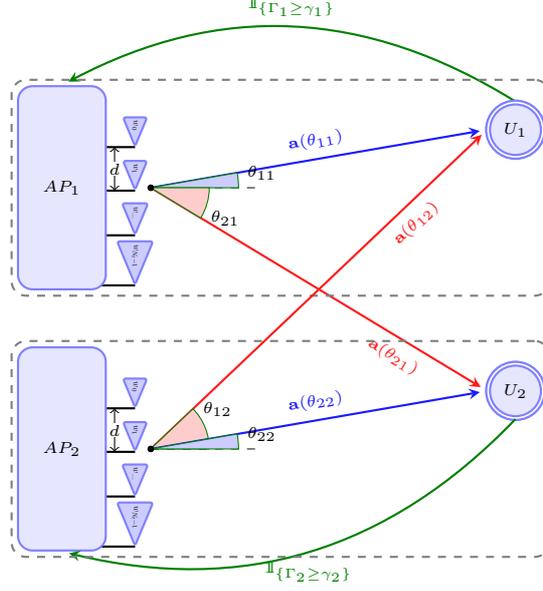
channel vectors arise when a uniform linear array (ULA) antenna is used at the transmitter under far-field, LoS conditions. Hence we can use the Vandermonde channel model as shown in equation (2.21).

$$\mathbf{a}_{ij} := \left[1, e^{-i\varphi_{ij}}, \dots, e^{-i\varphi_{ij}(N_t-1)} \right] \quad \forall i, j \in \mathcal{M} \quad (2.21)$$

Let φ_{ij} be defined as $\varphi_{ij} := \frac{2\pi d \sin(\theta_{ij})}{\lambda}$, and $i = \sqrt{-1}$. Where d is the antenna separation and λ is the wavelength of the carrier frequency f and $\theta_{ii} \neq \theta_{ij} \quad \forall i \neq j \in \mathcal{M}$.

The signal received by each receiver U_j is given as y_j ,

$$y_j = \underbrace{\mathbf{a}_{jj} \mathbf{w}_j s_j}_{\text{Useful Signal}} + \underbrace{\sum_{i \neq j} \mathbf{a}_{ij} \mathbf{w}_i s_i}_{\text{Interference Signal}} + \underbrace{n_j}_{\text{Noise}} \quad (2.22)$$


 Figure 2.6: System Model with $M = 2$

where \mathbf{w}_j is the pre-coder associated with the transmitter j , $\mathbf{a}_{jj}\mathbf{w}_j s_j$ is the useful signal for receiver j , $\sum_{i \neq j} \mathbf{a}_{ij}\mathbf{w}_i s_i$ is the interference coming from the transmitters $i \neq j$, and n_j represents the additive white noise with zero mean and σ_j^2 variance associated with the receiver j . s_j represent the symbol intended for user j with a symbol power of $|s_j|^2 = 1$. The symbols are assumed to be mutually uncorrelated. Based on the above description it is clear that the transmitter AP_j is operating in diversity mode i.e. only one symbol is transmitter per time slot.

Based on the above knowledge we can write the signal to interference and noise ratio (SINR) Γ_j of U_j as

$$\Gamma_j := \frac{|\mathbf{a}_{jj}\mathbf{w}_j|^2}{\sigma_j^2 + \sum_{i \neq j} |\mathbf{a}_{ij}\mathbf{w}_i|^2} \quad \forall i, j \in \mathcal{M}^2 \quad (2.23)$$

We consider a case with $M = 2$ transmit–receiver pairs where each transmitter is equipped with $N_t = 4$ transmit antennas. It is worth noting that this simple case of $M = 2$ is considered only for illustration clarity (in fact for this simple case we can find analytically a closed form expression of the Nash equilibrium and therefore we can compare the direction of transmission (DoT) obtained by our algorithm to this equilibrium), but obviously our scheme and

the analysis done so far in the thesis hold for larger but finite M . The values of the sinus perturbation parameters associated with equations (2.4) and (2.5) are given below. We summarize the system and simulation parameters in Table 2.3 and 2.4 respectively. Interested readers are referred to [FKB12b] for more details on how to select the parameters in Table 2.4.

Table 2.3: Summary of system parameters

d	$\frac{\lambda}{4}$
N_t	4
M	2
γ	10
ϵ_k	0.15
θ_{11}	0°
θ_{22}	0°
θ_{12}	-60°
θ_{21}	60°

Table 2.4: Summary of sinus perturbation parameters

user j	1	2
l_j	1.688	1.688
Ω_j	$2\pi 4.5185$	$2\pi 4.5425$
b_j	0.5	0.5
ϕ_j	0	0

For the above setting, we can show that Nash equilibrium is achieved if each AP_j transmits in a direction close to U_j i.e. to say $\psi_j \approx \theta_{jj} \forall j \in [1, 2]$. We run our algorithm by taking the initial values of $\psi_1^0 = -60^\circ, \psi_2^0 = 60^\circ$ which is the worst direction as it incurs strong interference on the receiver of the other network, and can cause outage. In Fig 2.7 we present the evolution of ψ_j and show that the transmitters direct their beams away from each other's receives and towards their own receiver which means that $\psi_j \rightarrow \theta_{jj} \forall j \in [1, 2]$ which is the Nash equilibrium point. Fig 2.7 shows that ψ_j convergences to a neighborhood of Nash since the step size ϵ_k used in the simulation is fixed (see Theorem 2).

It is worth mentioning that using fixed step size has the advantage of faster convergence rate as compared to vanishing step size at the expense of converging to a neighborhood of Nash (instead of converging to Nash). Comparison between fixed and vanishing step sizes is omitted due to space limitation.

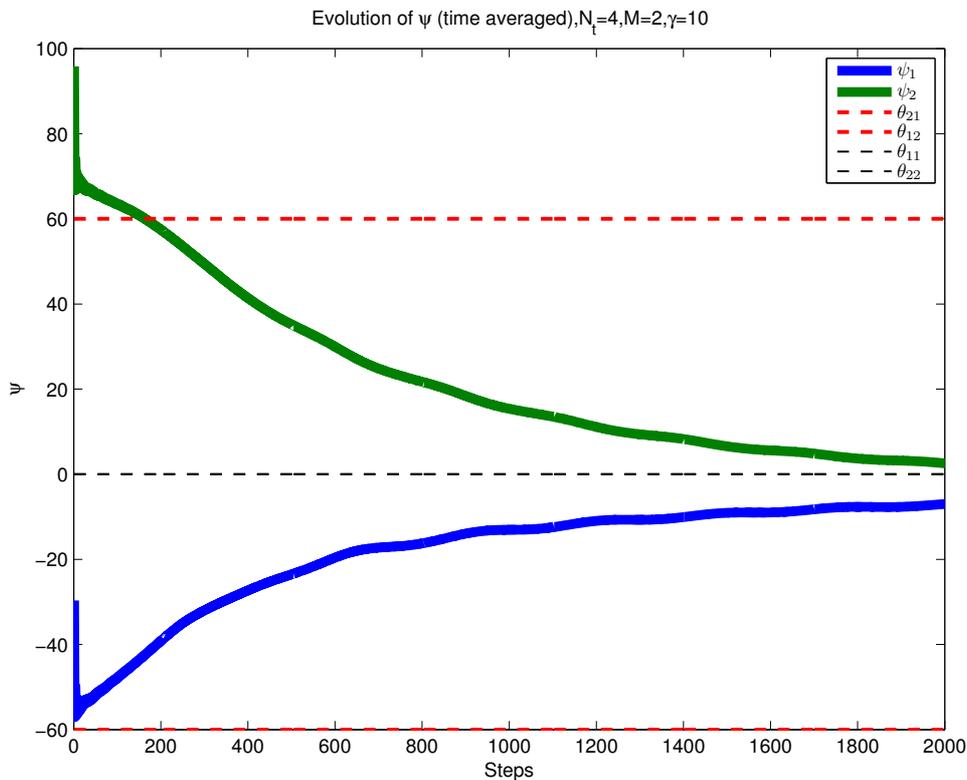
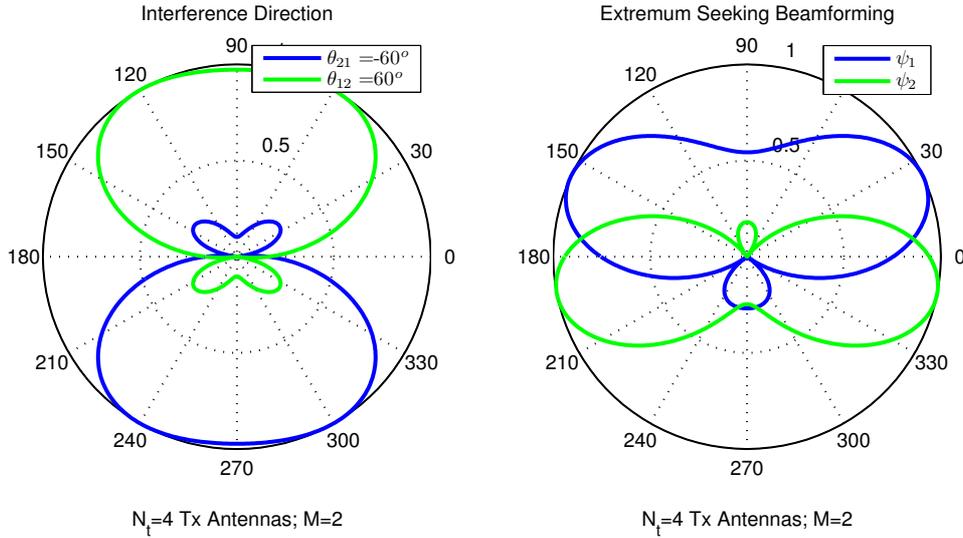


Figure 2.7: Evolution of ψ for AP_1 and AP_2

The scatter plots are presented in Fig 2.8 where the relative position of the interferers are presented on the left and the final direction of the beams after convergence are presented on the right.

2.6 Numerical Example C: Nash Bayesian Estimation

As the demand for wireless capacity is increasing, the cell sizes are shrinking to accommodate greater capacity and higher frequency reuse. Femtocells (FCs) are used to provide coverage and capacity to more users in a concentrated area using the shared wireless resources. Typically one or more FCs are deployed in an

Figure 2.8: Beam pattern for U_1 and U_2

existing macrocell (MC). A typical femtocell (FC) has a radius of 10 - 50 meters which is much smaller than the MC radius 300 - 2000 meters ([CA09, GKTH97]). Typically FCs use the same frequency as the macro cell which leads to several power control problems at the cell edge of the FC which is contained in the MC. FCs are user deployed and could be located randomly throughout a MC. Open access FCs allow MC users to access the network through them and closed access FCs don't allow macro users to have access to the network [XCA10]. In this thesis we will deal with closed access FCs. When a macro user located in the neighborhood of a FC is operating at the same frequency, it will receive strong interference from the FC users and femto access point (FAP) which results in significant degradation in performance. In such cases it is important to implement distributed power control that achieves a tradeoff between the quality of service (QoS) (or success probability) of femto users and that of macro users. We attempt to study such scenarios in this thesis. Some of the recent works dealing with femto-macro power control and interference management are described hereinafter. In [CA09], the uplink capacity and interference avoidance for two-tier FC networks are discussed. In [XCA10], Ping et al. compare open and closed access FCs in the uplink. In [IDME12], Ibrahim et al. present an Adaptive interference mitigation techniques for femto cells. Various approaches for enhancing autonomous power control at Femto under co-channel deployment of Macrocell

and Femtocell are presented in [YZ11]. In [HAT13] a robust distributed H^∞ based power control in a dynamic wireless network environment is presented by Hassan. A location based autonomous power control for LTE-A heterogeneous network is presented in [YW11]. Hong et al. in [HYC09] present a decentralized power control scheme in FC networks based on game theory. A game theoretic approach for an interference control problem with sub band scheduling is presented in [Ran10]. In [BHN11], Bendlin et al. develop probabilistic power control for heterogeneous cellular networks with closed-access FCs. In [WCLP12], Wu et al. provide a downlink outage probability analysis for co-channel femtocells in hierarchical 3-sector macrocells. In most of the existing works, the SINR distribution function is obtained (and therefore the outage) using some assumptions such as infinite number of interferers, poisson point process mobility model, etc. However, in a realistic scenario where the users mobility cannot be modeled by a poisson point process and where the environment is dynamic, an accurate closed form expression of the SINR density is hard to obtain. In addition, the interference depends on the decision of the other transmitters which makes the density characterization of the interference and therefore the SINR very challenging. Besides, in the standard power control framework [Yat95], the iterative power control (the power at time step k is updated using the SINR at time step $k - 1$) can achieve a target SINR under the assumption that the channel stays constant until the algorithm converges. In the current work, we assume that the wireless channel changes from one time step to another.

Figure 2.9 shows our system model. It is worth noting that although one user per femto (respectively per macro) is shown/used, our analysis holds for any number of macro and femto users and our analysis extension to this case is straightforward. It should be mentioned here for clarity that we are only considering the downlink scenario for femto and macro users. Let the macro access point (MAP) and the macro user (MU) be represented by subscript M, m respectively and the femto access point (FAP) and its femto user be represented by the subscript F, f respectively. Let h_{Mm} represents the channel between the MAP and the macro user, $h_{F_j f_i}$ represents the channel between the FAP j and the femto user i , and $h_{F_j m}$ represents the interference channel between the FAP j and the macro user. This channel $h_{F_j m}$ is typically unknown. $h_{M f_i}$ represents the channel between the MAP and the femto user i . Let $g_{Mm} \triangleq |h_{Mm}|^2$ be the channel gain between the MAP and the macro user. Similarly, we define

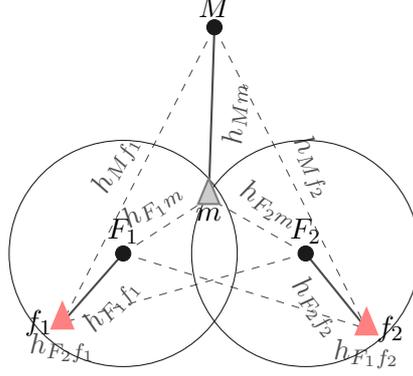


Figure 2.9: Macrocell containing two FCs

$g_{F_j f_i} \triangleq |h_{F_j f_i}|^2$, $g_{F_j f_i} \triangleq |h_{F_j f_i}|^2$, $g_{F_j m} \triangleq |h_{F_j m}|^2$ and $g_{M f_i} \triangleq |h_{M f_i}|^2$. Here we consider a situation where both FAPs and MAP are using the same frequency in the downlink and generating interference to each other's users.

The downlink SINR of macro user Γ_m is defined as

$$\Gamma_m \triangleq \frac{p_M g_{Mm}}{\sigma_m^2 + \sum_{i=1}^2 p_{F_i} g_{F_i m}} \quad (2.24)$$

Where p_M is the MAP power allocated to the macro user m . p_{F_i} is the power of the FAP i , σ_m^2 is the noise variance and $\sum_{i=1}^2 p_{F_i} g_{F_i m}$ represents the interference caused by the FAPs 1 and 2 on the macro user.

The downlink SINR of femto user i Γ_{f_i} is defined as

$$\Gamma_{f_i} \triangleq \frac{p_{F_i} g_{F_i f_i}}{\sigma_{f_i}^2 + \sum_{j \neq i} p_{F_j} g_{F_j f_i} + p_M g_{M f_i}} \quad (2.25)$$

Where $\sigma_{f_i}^2$ is the noise variance, $\sum_{j \neq i} p_{F_j} g_{F_j f_i}$ is the inter femto interference and $p_M g_{M f_i}$ represents the interference coming from the MAP. In addition, we assume that the MAP and FAPs have imperfect knowledge of their own users' SINRs. The observed SINR, respectively for macro and femto users, is therefore given as,

$$\widehat{\Gamma}_m^k = \Gamma_m^k + v_m^k \quad (2.26)$$

$$\widehat{\Gamma}_{f_i}^k = \Gamma_{f_i}^k + v_{f_i}^k \quad (2.27)$$

where v_m and v_{f_i} represent the respective errors due to feedback delays, channel estimation errors or quantization errors. To ensure that macro user is given priority while maintaining a reasonable quality of service (QoS) for the femto

users we formulate the following problem. Let the objective be to maximize the rewards R_m and R_{f_i} of macro and femto respectively (given respectively by equations (2.28) and (2.29)) where the rewards are constructed from the probabilities of success of femto and macro users. This type of rewards is more robust and ensures better QoS as compared to some existing works that maximize the expected SINR of each user.

$$R_m = \alpha \mathbb{E}_{\gamma_m} [\mathbb{P}(\Gamma_m \geq \gamma_m)] \quad (2.28)$$

$$R_{f_i} = \beta \mathbb{E}_{\gamma_{f_i}} [\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})] - \alpha (1 - \mathbb{E}_{\gamma_m} [\mathbb{P}(\Gamma_m \geq \gamma_m)]) \quad (2.29)$$

γ_m and γ_{f_i} are the target SINRs for respectively macro and femto users. $\alpha > 0$ and $\beta > 0$ are design parameters which can be adjusted to find the desired balance in performance between the femto and the macros. In the aforementioned formulation, the MAP tries to maximize the success probability of its own users without considering the presence of the FC users. It is the responsibility of the FAP to operate in such a way that it minimizes the interference to nearby macro users while trying to maintain a reasonable success probability for its own users. Notice also that the technique/analysis provided in the sequel still holds when the MAP tries also to obtain a trade-off between the success probabilities of the macro and femto users, i.e. when $R_m = \alpha \mathbb{E}_{\gamma_m} [\mathbb{P}(\Gamma_m \geq \gamma_m)] - \beta (1 - \mathbb{E}_{\gamma_{f_i}} [\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})])$. Obviously, our analysis/framework holds as well when the FAP attempts to maximize the success probability of its own users without considering the macro users (i.e. $R_{f_i} = \beta \mathbb{E}_{\gamma_{f_i}} [\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})]$). Furthermore, in the presence of multiple macro users, each user will have a reward function defined as in (2.28) and the objective of the macro cell in this case will be to maximize the sum of the rewards of all users (similar extension can be done in the case of multiple femto users). For clarity of presentation and ease of notation, we will limit ourselves to one user per femto (two FCs) and one macro user.

Furthermore, the targets γ_m and γ_{f_i} are determined by the network (e.g FAP, MAP) such that the signals are decoded with small Block Error Rate (BLER) and a minimum QoS is therefore ensured for users. It is worth noting that the target SINRs γ_m and γ_{f_i} may not be constant over time since the environment type is dynamic (e.g. the wireless channel type may change during the connection from an Outdoor Channel to an Indoor Channel and the target SINR must be

updated in this case). This dynamic adaptation of the target SINR explains the presence of $\mathbb{E}_\gamma[\cdot]$ in the rewards in equations (2.28) and (2.29) which represents the expectation over the values of target γ_m and γ_{f_j} attributed by the macro and femto access points (FAPs). In addition, it is evident that our framework holds when the target SINR is constant. We assume that the target γ_m is available to the FAP through a backhaul between MAP and FAP. In the absence of a backhaul between FAP and MAP, the FAP can listen to the downlink control channel of the MAP (since both are using the same technology/standard) where such information is usually sent. One can also assume that γ_m is broadcasted by the MAP and that the FAP can listen to the broadcast channel. It should be mentioned here that these targets γ_m and γ_{f_j} change slowly because they depend on the type of environment, as for example transition from outdoor to indoor or vice versa happens slowly. Notice that in cellular networks, the base station/AP updates the target SINR using the outer loop power control (called also slow loop power control as the update of the target SINR happens slowly) and informs the users of the updated target SINR using the downlink control channel. The update algorithm of target SINRs falls out of the scope of this work. Notice that in the case when the FAP tries to maximize the success probability of its own users without considering the macro users (i.e. $R_{f_i} = \beta \mathbb{E}_{\gamma_{f_i}} [\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})]$), there is no need to exchange the target γ_m between the MAP and the FAP.

Using the reward functions defined in equations (2.28) and (2.29), the distributed stochastic power control problem can be formulated as follows.

$$\max_{p_M, p_{F_1}, p_{F_2}} R_x \quad \forall x \in \{m, f_1, f_2\} \quad (2.30)$$

The success probabilities $\mathbb{P}(\Gamma_m \geq \gamma_m)$ and $\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})$ are given as,

$$\begin{aligned} \mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i}) &\triangleq \int_{\gamma_{f_i}}^{\infty} \Phi(\Gamma_{f_i}) d\Gamma_{f_i} \\ \mathbb{P}(\Gamma_m \geq \gamma_m) &\triangleq \int_{\gamma_m}^{\infty} \Phi(\Gamma_m) d\Gamma_m \end{aligned}$$

where $\Phi(\Gamma_m)$ and $\Phi(\Gamma_{f_i})$ are respectively macro and femto SINR densities. It is well known that Gaussian assumption of SINRs densities is not accurate since the SINR depends on various random parameters such as users' mobility, dynamic environment, wireless channel conditions as well as dynamic interference (that depends itself on the decision of other transmitters/users). Therefore the SINR density does not have an accurate closed form expression which makes

the distributed stochastic optimization problem in (2.30) very hard. The above problem cannot be solved using standard optimization techniques because of the complexity and randomness associated with such rewards. In order to solve the above problem, we first present a stochastic learning technique based on our recently developed theory in [HTAZ13e] that uses sinus perturbation and the instantaneous numerical value of the reward of to approach the problems defined in (2.30). As this techniques required the numerical value of the reward which is based on the density of SINR, we need to track the density $\Phi(\Gamma_m)$ and $\Phi(\Gamma_{f_i})$ using the imperfect instantaneous estimated SINR. Since the expression is too complicated, *only* numerical values of the rewards in (2.28) and (2.29) can be obtained/estimated at each time (due to some complicated integral expressions). Notice that, in order to obtain numerical values of the reward in (2.29), we exploit the fact that MC and FCs users are using the same standard and the FAP can listen to the uplink control channel of the macro user. For example, when the macro user who is at the cell edge of the FC transmits its SINR $\hat{\Gamma}_m$ to its MAP, the FAP can listen to this $\hat{\Gamma}_m$. However, we stress that this is necessary only because of the definition of the reward in (2.29). If instead of maximizing the reward in (2.29), the FAP tries to maximize the success probabilities of its own users (i.e. if $R_{f_i} = \beta \mathbb{E}_{\gamma_{f_i}}[\mathbb{P}(\Gamma_{f_i} \geq \gamma_{f_i})]$), the aforementioned assumption is not needed (i.e. the FAP does not have to listen to $\hat{\Gamma}_m$).

The problem in (2.30) can be solved using our Nash Seeking theory developed in [HTAZ13e] as follows. At each time instant k , each transmitter updates its power p_X^k (recall that X could be either M or F_i), by adding the sinus perturbation to the intermediary variable \hat{p}_X^k using equation (2.31), and transmits using p_X^k . Then, each transmitter (AP) computes a realization of the reward R_x^{k+1} at time $k+1$ which is used to compute the intermediary variable \hat{p}_X^{k+1} using equation (2.32). The power p_X^{k+1} is then updated using equation (2.31). This procedure is repeated for the whole transmission window (connection duration) $T > 0$.

The update is done in discrete time and is given by,

$$p_X^k = \hat{p}_X^k + a_X \sin(\Omega_X \hat{k} + \phi_X) \quad (2.31)$$

$$\hat{p}_X^{k+1} = \hat{p}_X^k + \lambda_k l_X a_X \sin(\Omega_X \hat{k} + \phi_X) R_x^{k+1} \quad (2.32)$$

where the grow rate $l_X > 0$ has a small value and the step size λ_k (also known as the learning rate) has also a small value $0 < \lambda_k < 1$ (e.g. $\lambda_k = \frac{1}{k+1}$). In addition, $\hat{k} := \sum_{k'=1}^k \lambda_{k'}$, and $\phi_X \in [0, 2\pi] \forall X \in \{M, F_1, F_2\}$. The condition $\Omega_X \neq \Omega_{X''}, \Omega_{X''} + \Omega_X \neq \Omega_{X'}$ need to be satisfied such that the perturbations

are orthogonal to ensure convergence of the algorithm. The above framework is only applicable to functions which are Lipschitz continuous and the existence of the equilibrium is assumed. One can refer to [HTAZ13e, FKB12b] for more details.

Bayesian Density Tracking

In order to use the extremum seeking framework provided in section 2.2.1 we need to estimate a numerical value of the reward which is based on the density of the SINR. In this section we use Bayesian Theory for SINR density tracking, we first model the SINR evolution as a dynamic state equation (not necessarily linear) as follows.

The transmit powers of the MAP and FAPs evolve according to the following equations.

$$\begin{aligned} p_M^{k+1} &= p_M^k + \eta_M^k, \\ p_{F_i}^{k+1} &= p_{F_i}^k + \eta_{F_i}^k. \end{aligned}$$

Where k represents the discrete time index and η_M^k and $\eta_{F_i}^k$ are the power deviation between times k and $k + 1$.

We can write the SINR of macro user at time $k + 1$ as

$$\Gamma_m^{k+1} = \frac{(p_M^k + \eta_M^k)g_{Mm}^{k+1}}{\sigma_m^2 + \sum_{i=1}^2 (p_{F_i}^k + \eta_{F_i}^k)g_{F_im}^{k+1}}$$

For simplification we define the variable $\hat{\epsilon}_m^k$

$$\hat{\epsilon}_m^k \triangleq \frac{\eta_M^k g_{Mm}^{k+1}}{\sigma_m^2 + \sum_{i=1}^2 (p_{F_i}^k + \eta_{F_i}^k)g_{F_im}^{k+1}}$$

The channel gain g_{Mm}^k evolves according to the following dynamic equation, $g_{Mm}^{k+1} = g_{Mm}^k + \varrho_{Mm}^k$ where ϱ_{Mm}^k is an ergodic stochastic process. Putting the value of $\hat{\epsilon}_m^k$ in the above equation we get.

$$\begin{aligned} \Gamma_m^{k+1} &= \frac{p_M^k g_{Mm}^{k+1}}{\sigma_m^2 + \sum_{i=1}^2 (p_{F_i}^k + \eta_{F_i}^k)g_{F_im}^{k+1}} + \hat{\epsilon}_m^k \\ \Gamma_m^{k+1} &= \frac{p_M^k g_{Mm}^{k+1}}{\left(\sigma_m^2 + \sum_{i=1}^2 p_{F_i}^k g_{F_im}^k\right) (1 + \epsilon_m^k)} + \hat{\epsilon}_m^k \end{aligned} \quad (2.33)$$

where

$$\epsilon_m^k \triangleq \frac{\sum_{i=1}^2 p_{F_i}^{k+1} \varrho_{F_i m}^k + \sum_{i=1}^2 \eta_{F_i}^k g_{F_i m}^{k+1}}{\sigma_m^2 + \sum_{i=1}^2 p_{F_i}^k g_{F_i m}^k}$$

In addition, we define ϵ_m^k as follows,

$$\epsilon_m^k \triangleq \frac{p_M^k \varrho_{Mm}^k}{\left(\sigma_m^2 + \sum_{i=1}^2 p_{F_i}^k g_{F_i m}^k\right) (1 + \epsilon_m^k)} + \epsilon_m^k$$

Using the above fact, equation (2.33) can be written as

$$\Gamma_m^{k+1} = \frac{\Gamma_m^k}{1 + \epsilon_m^k} + \epsilon_m^k$$

that implies,

$$\Gamma_m^{k+1} = \Gamma_m^k \sum_{l=0}^{\infty} (-\epsilon_m^k)^l + \epsilon_m^k \quad (2.34)$$

It is worth noting that in all iterative power control strategies (also in learning techniques) the state difference or power deviation between two consecutive iterations has to be small in order to ensure the convergence. Therefore, ϵ_m^k is small $\forall k$. Similarly it can be shown that the SINR Γ_{f_i} can be written as

$$\Gamma_{f_i}^{k+1} = \Gamma_{f_i}^k \sum_{l=0}^{\infty} (-\epsilon_{f_i}^k)^l + \epsilon_{f_i}^k \quad (2.35)$$

where $\epsilon_{f_i}^k$ is small $\forall k$. Note that ϵ_m^k and $\epsilon_{f_i}^k$ can be considered as random variables.

The SINRs of macro and femto users evolve according to equations (2.34) and (2.35), hence dynamic state equations and Bayesian theory can be used for density tracking.

For ease of notation and to conserve space we introduce the following SINR notation Γ_x^k , (where index x could be either m or f_i and X could be either M or F_i). We also use the notation $\Gamma_x^{k+1} = f_x^k(\Gamma_x^k, \epsilon_x^k)$ to represent equation (2.34) or (2.35).

We stress that only an imperfect measurement $\widehat{\Gamma}_x^k$ of Γ_x^k is available at time k . We use the notation $\widehat{\Gamma}_x^k = h_x^k(\Gamma_x^k, v_x^k)$ to represent equation (2.26) or (2.27). We thus define D_x^k as the set of previous imperfect SINR measurements till time k , i.e. $D_x^k = \{\widehat{\Gamma}_x^i : i = 1, \dots, k\}$. We also denote by $\Phi(\Gamma_x^k | D_x^{k-1})$ the SINR density estimation assuming that the set of measurements D_x^{k-1} is available.

Using recursive Bayesian estimation from [GSS93], we can track the SINR density $\Phi(\Gamma_x^k|D_x^k)$ as follows. First, we find the density $\Phi(\Gamma_x^k|D_x^{k-1})$ and then we estimate the required density $\Phi(\Gamma_x^k|D_x^k)$.

$\Phi(\Gamma_x^k|D_x^{k-1})$ can be obtained using the following equation (2.36) from [GSS93],

$$\Phi(\Gamma_x^k|D_x^{k-1}) = \int \Phi(\Gamma_x^k|\Gamma_x^{k-1})\Phi(\Gamma_x^{k-1}|D_x^{k-1})d\Gamma_x^{k-1} \quad (2.36)$$

where

$$\Phi(\Gamma_x^k|\Gamma_x^{k-1}) = \int \Phi(\Gamma_x^k|\Gamma_x^{k-1}, \epsilon_x^{k-1})\Phi(\epsilon_x^{k-1}|\Gamma_x^{k-1})d\epsilon_x^{k-1}$$

using the fact that $\Phi(\epsilon_x^{k-1}|\Gamma_x^{k-1}) = \Phi(\epsilon_x^{k-1})$ we have

$$\Phi(\Gamma_x^k|\Gamma_x^{k-1}) = \int \delta(\Gamma_x^k - f_x^{k-1}(\Gamma_x^{k-1}, \epsilon_x^{k-1}))\Phi(\epsilon_x^{k-1})d\epsilon_x^{k-1} \quad (2.37)$$

Where $\delta(\cdot)$ is a dirac function. The dirac function arises because when Γ_x^{k-1} and ϵ_x^{k-1} are known, Γ_x^k is obtained from equation (2.34) or (2.35); i.e. $\Gamma_x^k = f_x^{k-1}(\Gamma_x^{k-1}, \epsilon_x^{k-1})$. Now we are able to estimate the density $\Phi(\Gamma_x^k|D_x^{k-1})$ by putting the value of $\Phi(\Gamma_x^k|\Gamma_x^{k-1})$ from equation (2.37) into equation (2.36).

At time step k , the set of measurements D_x^k becomes available. Using the Bayes' rule, we can update the SINR density $\Phi(\Gamma_x^k|D_x^k)$ using the following equation,

$$\Phi(\Gamma_x^k|D_x^k) = \frac{\Phi(\hat{\Gamma}_x^k|\Gamma_x^k)\Phi(\Gamma_x^k|D_x^{k-1})}{\Phi(\hat{\Gamma}_x^k|D_x^{k-1})} \quad (2.38)$$

where

$$\begin{aligned} \Phi(\hat{\Gamma}_x^k|D_x^{k-1}) &= \int \Phi(\hat{\Gamma}_x^k|\Gamma_x^k)\Phi(\Gamma_x^k|D_x^{k-1})d\Gamma_x^k \\ \Phi(\hat{\Gamma}_x^k|\Gamma_x^k) &= \int \delta(\hat{\Gamma}_x^k - h_x^k(\Gamma_x^k, v_x^k))\Phi(v_x^k)dv_x^k \\ \Phi(\Gamma_x^1|D_x^0) &\equiv \Phi(\Gamma_x^1) \end{aligned}$$

Joint Density Tracking and Power Control Algorithm

By combining the techniques described in section 2.2.1 and 2.6 we get the following density tracking and power control algorithm (see Algorithm 3 below). First each users initializes $\Phi(\Gamma_x^1|D_x^0)$ with a distribution $\Phi(\Gamma_x^1)$, then each access point (AP) X initializes its power and transmits. After this the loop begins. At

time k each AP observes $\widehat{\Gamma}_m^k$ and $\widehat{\Gamma}_{f_i}^k$ which are the noisy versions of the SINRs and updates their previous density according to equation (2.38). Then each AP estimates its reward R_x^k from equation (2.28) or (2.29), then this reward is used to calculate the power according to (2.31). This power is used for making the transmission. This process is repeated until horizon T (which is the connection duration). In the next section we present the numerical results which show

- 1: At $k = 1$ initialize $\Phi(\Gamma_x^1|D_x^0) \equiv \Phi(\Gamma_x^1)$
- 2: Each AP X , initialize p_X^0 and transmit
- 3: Repeat
- 4: Each AP X at time k , observes $\widehat{\Gamma}_m^k$ and $\widehat{\Gamma}_{f_i}^k$
- 5: Update the previous density according to equation (2.38)
- 6: Evaluate the reward R_x^k from equation (2.28) or (2.29)
- 7: Calculate power according Equation (2.31)
- 8: Update \hat{p}_X^{k+1} using Equation (2.32)
- 9: until horizon T

Algorithm 3: Recursive Bayesian Power Control Algorithm (Recursive B-PCA)

that our algorithm performs better than existing methods. We used the following settings for simulating the above framework. The macro cell radius is equal to 500m whereas the radius of each femto is 25m. We consider a frequency selective Rayleigh fading channel with exponential delay profile. The power spectral density of noise is -174 dBm/Hz. The path losses are calculated according to Cost-Hata Model. In addition, we used the following sinus framework parameters in the simulations. $l_{f_1} = 2.5; l_{f_2} = 3.5; l_m = 4.5; \phi_{f_1} = 0; \phi_{f_2} = 0; \phi_m = 0; \Omega_{f_1} = 1.4; \Omega_{f_2} = 1.2; \Omega_m = 1; b_{f_1} = 0.6; b_{f_2} = 0.3; b_m = 0.1; \alpha_m = 1; \beta_{f_i} = 3$. Interested readers are referred to [FKB12b] for more details on how to select the above parameters. In addition, our results were generated using $\lambda_k = \frac{1}{k+1}$ (since we proved in [HTAZ13e] the convergence to Nash equilibrium for $\lambda_k = \frac{1}{k+1}$). The results for success probability are presented in Figure 2.10 where we compare our scheme to the widely used standard power control scheme from [Yat95]. Figure 2.10 shows that, with our proposed scheme, both femtos and macro cells are able to maintain a success probability of 65 – 75% which is much better in comparison to the Yates method [Yat95]. It should be noted here that achieving a success probability of 65 – 75% for macro and femto users is quite significant

considering that we are dealing with fully distributed stochastic system with interacting player/APs, using only imperfect SINR. Please also refer to Figure 2.11 which shows the Pareto boundary for our conditions. A Pareto plot is generated by using all possible transmit power scenarios by all the APs, and it is desirable to be as close to the boundary as possible as that gives the best performance for all users. The white circle in Figure 2.11 represents our operating region which is quite close to the boundary. It is evident that achieving higher success probability for macro and femto users simultaneously is quite difficult in a stochastic environment.

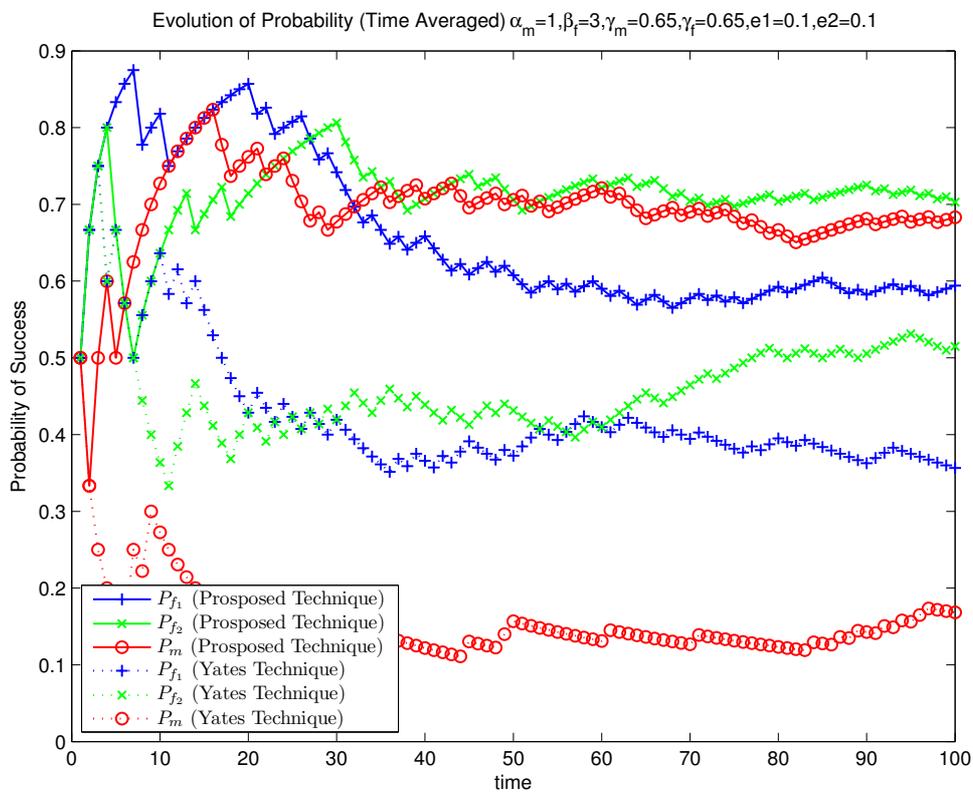


Figure 2.10: Evolution of Probability of Macro and Femto Users

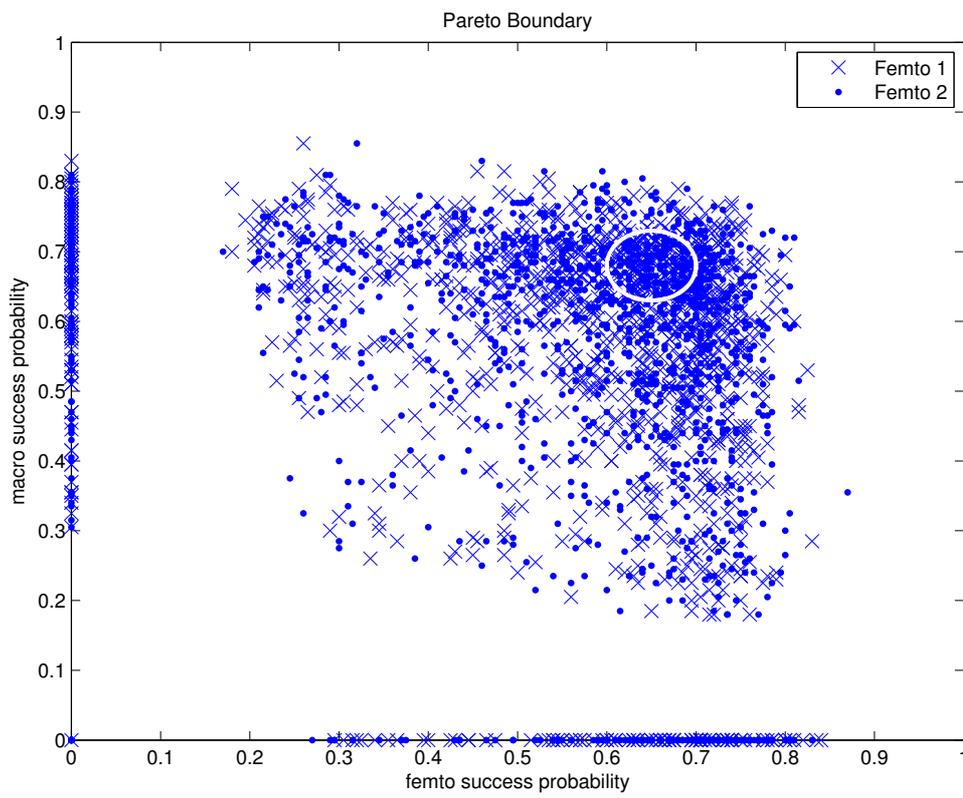


Figure 2.11: Pareto Boundary

Chapter 3

Resource Sharing with Mean Field in Cloud Networks

3.1 Introduction

Cloud-based applications transfer more and more data (sometimes even with real-time requirements). This strong growth and continual changes in the traffic characteristics and usage behavior raises questions on how to share limited capacity resources fairly and more efficiently. In order to get a certain level of fairness between users, the payoff function should capture a fair sharing policy. Foster et al.(2008) defines the ambiguous cloud as “A large-scale of distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet” [FZRL08]. The concept of cloud networking has emerged to satisfy the uneven demand of users wanting cheap computing resources ranging from storage to processing. Every aspect of a computer system can essentially be offered as a service to various users. Thanks to advances in virtualization and scheduling techniques it is possible to have multiple instances of a software running on the same physical system.

Resource sharing solutions are very important for data centers as it is required and implemented at different layers of the cloud network. The resource sharing problem can be formulated as a strategic decision-making problem between the competing players. Cloud computing has been gaining momentum and currently many big companies such as Amazon, Google and Microsoft are offering various

cloud computing and storage services.

Although there are many types of cloud based resources that can be shared over the internet, here we only discuss computing resources. Consider a scenario where an entity requires high computing resources once every month. Now they can rent computing resources on demand instead of physically owning dedicated computing resources. With the advent of modern operating systems and advances in networking technologies and virtual machine (VM) technologies it is now possible to host multiple instances with limited number of physical hardware. Amazon has been among the first companies to convert this into a viable business with the ‘EC2 service’. ‘Compute Engine by Google’ and ‘Windows Azure by Microsoft’ are the other major players in computing resources. These services are mostly used by companies and research organizations.

There are various cloud computing paradigms which are listed below. “SPI is an acronym for the most common cloud computing service models, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Software as a Service (SaaS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the internet. Platform as a Service (PaaS) is a paradigm for delivering operating systems and associated services over the Internet without downloads or installation. Infrastructure as a Service (IaaS) involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components. The increasing selection of services delivered over the Internet is sometimes referred to as XaaS, where XaaS refers to anything as a service.” [Rou12]. In IaaS cloud computing paradigm a virtual machine (VM) is offered to paying clients as a service. A virtual machine as the name implies is a virtual computing terminal hosted over physical servers. Virtual machines are created on demand and are able to satisfy client demand for a remotely accessible machine with a give set of hardware specifications. The VM has certain hardware specifications such as a CPU clock rate RAM and available storage space, which defines a certain minimum performance criterion. Chip Microprocessor (CMP) architecture is a scalable and cost effective architecture where a could comprises of a large number of commodity hardware which is relatively cheap as compared to having specialized hardware. The VM reside on top of chip microprocessor architecture (CMP) which enables them to be scalable by using shared resources. There is contention between the VMs as

they are using these shared physical resources like storage space, CPU processing power and last level cache (LLC). As all these machines have their respective services level agreement (SLA) i.e. minimum performance criterion that needs to be ensured. The resource contention could cause violation of the SLA of individual VMs as more virtual machines are hosted by a given number of physical machines. An example of resource contention at the cache level is the following. For example a Sun microprocessor with a 3MB Level 2 cache can share this cache among 32 different threads of execution. These different threads could belong to instances of 32 different VMs, in which case the resource management needs to be done in an efficient manner such that the SLA for each individual user is satisfied. There can be similar examples in the context of storage space and available CPU processing time for each VM.

So it is very important to manage these various resources in an efficient manner. This concept of contention in cloud based resources can be modeled using the concept of game theory. Following are some related works where game theory has been used in the context of cloud computing. In such a situation, but in a different context, Tullock (1980, [Tul80]) proposed a game-theoretic setting with probabilistic reward in *rent-seeking*. He has noticed earlier that the social loss associated to rent-seeking behavior exceeds a certain threshold. From a cloud networking perspective, this means that lot of resources may be wasted if the VM consider an economic renting. In this chapter, we study a stylized model of resource sharing in a distributed and fair manner. We will show that efficiency can be significantly improved for large cloud networks using a suitable price design. We examine both stability and myopic reaction using mean-field-taking strategies [AFA05]. Mean field games are known to be well-adapted for both finite (but large) and infinite systems Please refer to [GLL11] for an introductory reference on mean field games. In contrast to the other classical tools for large-scale systems, the mean-field approach incorporates the dynamics which allows online demand management (pay-as-you-use scheme). It allows optimization, control and dynamic mechanism design when the number of active users varies and studies the interactive nature of cloud networking, leading to strategic decision-making problems. Recent studies on cloud networking have already adopted a mean-field regime.

3.1.1 Overview

Game-theoretic approaches to cloud networking

The concept of cloud networking is very old but it has recently become a reality. There are several aspects of cloud networking that still need to be discussed. Game-theoretic tools for cloud networking have been recently proposed in [VJ10]. The authors in [VJ10] studied cloud resource allocation games. The work in [WVZX10] presents a game-theoretic method of fair resource allocation for cloud services. They proved the existence of a Nash equilibrium under feasibility condition of the resource allocation problem. [MCL⁺10] examines a resource allocation for VMs using Kelly's approach. The authors in [KDF⁺11] has studied optimal service pricing for cloud cache services. [ABYST12] studied the framework of Resource-as-a-Service (RaaS). In [KA06] a non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation are examined. [RY12] proposed a game-theoretic approach for the provisioning and operation of the infrastructure under uniform cost models. The authors in [KK12] tried to answer the question whether or not to utilize the cloud for processing, by identifying characteristics of potential cloud beneficiaries and advisable actions to actually gain financial benefits. They proposed a game-theoretic model of an Infrastructure-as-a-Service (IaaS) cloud market, covering dynamics of pricing and usage. The work in [APP11] modeled a service provisioning problem in cloud as an action-coupling constrained game, and proposed an efficient algorithm for the run time management and allocation of IaaS resources to competing SaaS (Software as a Service). The authors in [TM10] proposed a Bayesian Nash Equilibrium Allocation algorithm to solve resource management problem in cloud networking under heterogeneous distribution of resources and incomplete information. The concept of heavy traffic approximation of equilibria in resource sharing games is studied in [WBJ12].

Our work differs from the above references in three ways. First, the above reference do not examine the stability of the solution concepts. Second, their approaches seem not to be adapted to large-scale systems. Third, the references [RY12, KK12, APP11, TM10, WVZX10, KA06] do not propose simple, memory-less and easy-to-implement learning algorithms. In this chapter we will address these three issues.

Large-scale clouds

Large data and large number of users are frequently met in the context of cloud networking, such as in Amazon's Elastic Compute Cloud (Amazon EC2) which is a web service that provides resizable computing capacity in the cloud. As systems become large traditional game theoretic analysis result in computationally inefficient frameworks as they take into account effect of each unique action by every player. Recently, there has been renewed interest in developing large-scale strategic learning also called mean-field learning. These frameworks are very efficient at simplifying the system structure giving us few parameters such as the mean instead of tracking the unique action of each user. This not only helps in computationally when simulating or analyzing these systems, but it also drastically simplifies the mathematical analysis giving us the averaged effect of the interaction of a large number of particles instead of tracking each particle. [AJ10, AAJW07, Hua12] Thus, mean-field learning is an important framework for large-scale cloud networks. Such a problem is tackled in [PB11a] using block splitting for large-scale distributed learning. We refer the reader to Chapter 1 of [Tem12] for a recent survey book on distributed strategic learning.

3.1.2 Contribution

Although here we consider cloud networks, our resulting analysis is more general and is applicable to similar resources sharing networks. The contribution of this chapter can be divided into two main parts: i) Equilibrium analysis of two different resource sharing problems for finite and infinite number of clients and ii) development of low complexity distributed learning algorithms that can achieve the equilibrium solutions. The equilibria analysis shows the interest of using mean field theory in the cloud as on one hand it simplifies hugely the analysis of the game and allows thus developing low complexity iterative learning solutions that converge to the equilibrium (see section 3.4.2). Our contributions in this chapter are listed below:

- We first consider a cloud resource sharing problem with continuous action spaces where each user tries to maximize its own payoff. The payoff depends on the fraction of resources (e.g. LLC memory, CPU power, etc.) allocated to the user. This problem can be applied to the cases where each user tries to get VMs with resources as much high as possible in order

to reduce for example the computation time of his own tasks. This can be applied to the case where the cloud does not have enough physical resources to satisfy all the current clients' demands (especially for very high number of users). In the finite regime (finite number of users), we provide closed-form expressions of three important game-theoretic solution concepts: Nash equilibrium (NE), finite myopic mean-field response (F-MFR) and finite evolutionary stable strategy (F-ESS). The first surprising result is that the Nash equilibrium is not an evolutionary stable strategy for finite population (see Proposition 2). In fact in cloud networks, the number of users as well as users' demands are constantly changing and in this context evolutionary stable strategy (i.e. it cannot be invaded by small fraction of deviants) is crucial. Based on the work of Possajennikov (2003, [Pos03]) we show that mean-field-taking strategies are evolutionary stable strategies for a certain range of return index (i.e. α). We provide the explicit optimal pricing as a function of the number of active users. Under the optimal pricing no resource is wasted at the equilibrium. This means that the efficiency loss tends to zero.

- We then consider the infinite regime, because in many cloud networks the number of users is expected to be very high. We provide closed-form expression of the mean-field equilibrium and show how the mean field theory simplifies the analysis in this regime. Furthermore, we show that the mean-field equilibrium is an evolutionary stable strategy, i.e., it cannot be invaded by a small fraction of deviants. We show that the mean-field equilibrium is evolutionary stable for any fraction of deviants strictly less than 100% which makes our result even stronger. We also show that Nash equilibrium is stable in the infinite regime. This is due to the fact that in the limiting case, the influence of one generic player on the total demand is negligible.
- We then consider another resource sharing problem in the cloud where each user tries to satisfy anytime anywhere its own request (for example when enough physical resources are available for all users). The problem is formulated as a satisfactory game. In the finite regime, we provide a closed form expression of the satisfactory solution and develop a distributed algorithm which is able to reach it. Contrary to the aforementioned equilibrium concepts, the extension of the obtained satisfactory solutions to the infinite

regime is straightforward.

- We then provide distributed iterative learning algorithms that converges to the equilibrium solutions of the first resource sharing problem discussed above. for the finite regime, we provide an Ishikawa-based distributed iterative learning algorithm that converges to each of the three aforementioned equilibrium concepts and discuss its convergence time. It is worth mentioning that developing learning algorithms with less information and having faster convergence time is of high interest. We therefore propose a model-based (but still with less information) learning scheme for games with continuous action space and large number of players. Each player will update its learning strategy based on an aggregative term, which is the mean of an increasing function of the action of the other players. Each player will be influenced by the aggregate, and the mean-field behavior is formed from the action of each player. Each player will try to conjecture the aggregative term consisting of the actions of the other players at each time slot, and will respond to the aggregative term locally. This drastically simplifies the dimensionality of the mean-field response system in the asymptotic case. We also discuss the convergence time of the proposed learning algorithm and show that it is faster than the Ishikawa-based algorithm or standard contraction mapping fixed-point algorithms.

3.1.3 Structure

The remainder of the paper is organized as follows. In section 3.2 we present the resource sharing and the demand satisfaction framework. In Section 3.3 we examine both equilibrium and stability properties of the game with finite number of VMs and asymptotic game with infinite number of VMs. In section 3.4 the algorithms used to approach the equilibria are presented. In section 3.5 numerical results are presented.

3.2 Model

We consider resource (i.e. last level cache (LLC)) sharing between n virtual machine (VM) instances. These VMs, maximize their performance by increasing their access to the shared LLC. These VMs are trying to share the total available cache capacity given by c_n among themselves in a fair fashion. It is interesting to

analyze this type of problem in the context of game theory. As there is contention between various instances of the VM, each of these VM can be considered a player in a game competing for the same resources. Notice also that the physical resources in the cloud are not centralized (i.e. in one machine) which justify the need for a game framework.

There are several types of problems or scenarios that arise in this context. We focus here on two resource sharing problems. The first problem corresponds for example to the case where there are not enough resources of cache available to satisfy all clients or players. In such scenario we define a reward function which is based on a resource sharing strategy where each client demands a percentage of the total available cache capacity c_n and pays a price for this access. This ensures that there is fair sharing of the resources among the clients. The remaining demand of each client is met by secondary sources which are relatively more expensive. This first problem can arise also when there are enough resources for all clients and the clients try to get as much resources as possible in order to increase the performance of their VMs (e.g. in order to minimize the computation time of the required tasks). In the next two sections (3.2.1 and 3.2.2) we shall describe the resource sharing game for the finite and the infinite number of clients for such a scenario. The second scenario described in section 3.2.3 represents the case when the total available cache capacity is higher than the total demand and each client wants to use a given percentage of the total capacity/resources (satisfaction game).

3.2.1 Resource sharing game \mathcal{G}_n with finite number of clients

We consider resource sharing game with finite (but arbitrary) number of clients in a cloud network. We denote such a game by \mathcal{G}_n , where n is the number of clients. The action space of every client is $\mathcal{A} = \mathbb{R}_+$ which is a convex set, i.e., each player j chooses an action a_j that belongs to the set \mathcal{A} . An action may represent a certain demand for a resource. In case of cloud computing the demand could be the CPU access duration or the share of a percentage of total LLC memory. For example in case of a computation intensive task where several VMs are running simulation related computation tasks and want to complete their computation tasks as soon as possible by demanding for as much resources as possible. All the actions together determine an outcome as the resources are shared. Let p_n be the unit price of cloud resource usage by the clients. Then, the payoff of player

j is given by

$$r_j(a_1, \dots, a_n) = c_n \frac{h(a_j)}{\sum_{i=1}^n h(a_i)} - p_n a_j, \quad (3.1)$$

if $\sum_{i=1}^n h(a_i) > 0$ and zero otherwise. The structure of the reward function $r_j(a_1, \dots, a_n)$ for client j shows that it is a percentage of allocated capacity minus the cost for using that capacity. Here, c_n represents the value of the available resources (which can be seen as the capacity of the cloud), h is a positive and nondecreasing function with $h(0) = 0$. We fix the function h to be x^α where $\alpha > 0$ denotes a certain return index. The function h represents hence the efficiency of the demand in the access probability to the resource. This introduction of efficiency can be interpreted by the fact that the clients and the physical resources are distributed and are connected through internet or virtual private networks. This class of games have been studied by Tullock in the 1980s under the name of rent-seeking game. See Tullock (1980, [Tul80]). However [Tul80] does not examine stability, price design and algorithmic issues. The cloud game \mathcal{G}_n is given by the collection $(\mathcal{N}, \mathcal{A}, (r_j)_{j \in \mathcal{N}})$ where $\mathcal{N} = \{1, \dots, n\}$, $n \geq 2$, is the number of potential players. We say that player j is *active* if its action $a_j > 0$. We will see that what is important in our analysis is not the number of potential players \mathcal{N} but the set of active players (those with non-zero demand). Therefore, we define the following variable:

$$m_n \triangleq \left(\frac{1}{n} \sum_{i=1}^n a_i^\alpha \right)^{\frac{1}{\alpha}}$$

as a weighted α -norm of all the actions. m_n will play the role of the mean-field term in the resource sharing game and will be useful in the development of iterative algorithm but also in the study of mean-field games. The payoff function can be rewritten as $\bar{r}_n(a_j, m_{n,-j}) \triangleq r_j(a) = \frac{c_n}{n} \left(\frac{a_j}{m_n} \right)^\alpha - p_n a_j$, where the ratio $\frac{c_n}{n}$ is the theoretical capacity per client in the cloud and

$$m_{n,-j}^\alpha \triangleq \frac{1}{n-1} \sum_{i \neq j} a_i^\alpha = \frac{n}{n-1} \left(m_n^\alpha - \frac{a_j^\alpha}{n} \right)$$

$m_{n,-j}$ represents the mean when the action of user j is not taken into account.

$$\bar{r}_n(a_j, m_{n,-j}) = \left(\frac{c_n a_j^\alpha}{m_{n,-j}(n-1) + a_j^\alpha} \right) - p_n a_j,$$

The payoff function of a generic player j depends only on own-action a_j and the mean m_n . Note that $m_{n,-j}^\alpha$ does not depend on a_j .

Remark 3. *The payoff is discontinuous for $\sum_i h(a_i) = 0$. As we will see in the next sections the discontinuity of the payoff function (at the origin) can be handled in the equilibrium analysis.*

As mentioned in [MCL⁺10], the above payoff function has interesting connections with the seminal works of Kelly on capacity sharing problems. However, the approach presented here has additional features:

- (i) our approach is distributed and autonomous decision approach.*
- (ii) the evolutionary stability of the resulting system depends on the interactive and strategic behavior of each client,*
- (iii) Our payoff is well adapted even if there is no coordination between the clients for joint decision-making,*
- (iv) The mean-field approach and the evolutionary stable strategies of the finite game presented here complement the previous works.*

3.2.2 Resource sharing game \mathcal{G}_∞ with infinite number of clients

In current cloud network, the number of clients for clusters grows without bound and it is becoming increasingly important to dynamically share resources between these large population of clients. When the number of active clients is very large, we analyze the asymptotic game, denoted by, $\mathcal{G}_\infty = (\mathcal{A}, \bar{r})$ where the payoff denoted by \bar{r} can be obtained easily as follows (using the expressions of m_n and $\bar{r}(a_j, m_{n,-j})$ defined above in the finite regime and then taking the limit when n tends to infinity) $\bar{r} : \mathcal{A}^2 \rightarrow \mathbb{R}$,

$$\bar{r}(a_j, m) = \begin{cases} \frac{ca_j^\alpha}{m^\alpha} - pa_j, & m > 0 \\ 0, & m = 0, a_j = 0 \end{cases}$$

The numbers $c > 0$ and $p > 0$ are scaled asymptotic version of $\frac{c_n}{n}$ and p_n i.e. $c = \lim_{n \rightarrow +\infty} \frac{c_n}{n}$ and $p = \lim_{n \rightarrow +\infty} p_n$. The term m can be interpreted as a limit value of m_n i.e. $m = \lim_{n \rightarrow +\infty} m_n$. Therefore, we can write $\bar{r}(a_j, m) = p \left(\frac{c}{p} \frac{a_j^\alpha}{m^\alpha} - a_j \right)$. As we will see, one of the important terms in the equilibrium structure will be the ratio $\frac{c}{p}$ which can be interpreted as the inverse of the *cost (price) per capacity*.

We will show in the next section that the error gap between the mean-field game model and the finite game model is the error at which $\frac{c_n}{n} \frac{1}{p_n}$ goes to $\frac{c}{p}$ together with a convergence rate of $O(\frac{1}{n})$. This result is interesting because when the system is large, one can approximate it by a very simple mean-field

game model for any number of active clients that exceeds n_ϵ in order to get an error that is at most ϵ . In addition the mean-field game system has additional features: it provides

- A stable equilibrium that is resilient to small perturbations.
- The mean-field equilibrium is robust to perturbation by new entrants/exits in the sense that it remains within ϵ -equilibrium.
- The payoff at mean-field equilibrium is close to the payoff of all the finite games for any active number of clients above a certain threshold. This means that mean-field game model is also useful in finite regime (even at small scale).

3.2.3 Demand satisfaction game \mathcal{G}_s

In this subsection we analyze a different type of game where each client is interested in satisfying a given demand instead of trying to maximize a given reward function. First we provide the characterization of an equilibrium for finite clients and then we present a fully distributed algorithm which is able to reach this equilibrium. It is straight forward to extend the equilibrium characterization to the infinite case as it is shown at the end of this section. We restrict the set \mathcal{A} to the interval $[0, c_n]$.

We now examine learning satisfactory solutions in cloud resource sharing games with *continuous action space*. Our motivations for satisfactory solution seeking are the following: First, we observe that in dynamic interactive system, most clients constantly make decisions which are simply “good enough” rather than best response or optimal. Simon (1956, [Sim56]) has adopted the word “satisficing” for this type of decision. As mentioned by Simon himself in his paper in page 129, “*Evidently, organisms adapt well enough to ‘satisfice’; they do not, in general, ‘optimize’.*” Therefore satisfactory solution offers an alternative approach and is closely model the way VMs make decisions [Sti03, GSF98, SG99].

Here, a satisfactory strategy is a decision-making strategy that attempts to meet an acceptability threshold. This is contrasted with optimal decision-making or best response strategy, an approach that specifically attempts to find the best option available given the choice of the other clients. Following that idea we define a satisfaction solution as a situation where every client is satisfied, i.e., its payoff is above its satisfactory level.

For each client j , we introduce a satisfaction level s_j^* . The collection $\mathcal{G}_s \triangleq (\mathcal{N}, (\mathcal{A}, s_j, s_j^*)_{j \in \mathcal{N}})$ constitutes a one-shot game in satisfaction form.

Definition 5 (Demand Profile). *The demand profile $(a_j)_{j \in \mathcal{N}}$ is a pure satisfactory solution of the cloud resource sharing game if all the clients are satisfied:*

$$s_j(a) = \frac{c_n a_j^\alpha}{\sum_{i=1}^n a_i^\alpha} \geq s_j^*, \quad \forall j \in \mathcal{N}.$$

The above inequality implies that the satisfaction $s_j(a)$ represents a fraction of the total capacity allocated to client j which should be greater or equal to s_j^* .

3.3 Equilibrium Analysis

In order to study the cloud in a dynamic and constantly changing setting, we examine the stability concept in a finite game \mathcal{G}_n . We first start by characterizing the Nash equilibrium and then we compare their equilibrium/strategy to the finite evolutionary stable strategy (F-ESS). We will see that Nash and F-ESS have two different equilibria (see proposition 2). Which means that Nash equilibrium is not stable.

We then consider another strategy called finite myopic mean-field response (F-MFR) and show that this strategy is stable under the system model considered in this chapter. In the last part of this section, we develop a distributed iterative algorithm that can converge to each of these three equilibria studied in this section.

3.3.1 Equilibrium for \mathcal{G}_s finite number of clients

Before defining Nash equilibrium, we introduce the so-called reaction set or best response correspondence. The best response problem (BR_j) of player j is

$$(BR_j) : \text{Given } a_{-j} \triangleq (a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n),$$

$$\text{Find } a_j \in \arg \max_{a'_j} r_j(a'_j, a_{-j})$$

Which is the best response of client j given the demand of the others.

An interior solution to the problem (BR_j) (whenever it exists) is obtained by setting the first order condition (FOC) to zero and the second order derivative to be negative.

Note that the payoff is differentiable outside the origin and the first derivative with respect to a_j yields

$$\begin{aligned}\partial_{a_j} \bar{r}(a_j, m_n) &= \frac{c_n}{n} \left[\frac{\alpha a_j^{\alpha-1} m_n^\alpha - a_j^\alpha \left(\frac{\alpha}{n} a_j^{\alpha-1} \right)}{m_n^{2\alpha}} \right] - p_n \\ &= \frac{c_n \alpha a_j^{\alpha-1}}{n} \left[\frac{m_n^\alpha - \frac{a_j^\alpha}{n}}{m_n^{2\alpha}} \right] - p_n.\end{aligned}$$

and using the fact that $m_n^\alpha - \frac{a_j^\alpha}{n} = \sum_{i \neq j} \frac{a_i^\alpha}{n}$ is independent of a_j we get the second derivative $\partial_{a_j}^2 \bar{r}(a_j, m_n)$ as

$$\begin{aligned}&\triangleq \frac{c_n \alpha}{n} \left(m_n^\alpha - \frac{a_j^\alpha}{n} \right) \left[\frac{(\alpha-1) a_j^{\alpha-2} m_n^{2\alpha} - a_j^{\alpha-1} \frac{2\alpha}{n} a_j^{\alpha-1} m_n^\alpha}{m_n^{4\alpha}} \right] \\ &= \frac{c_n \alpha}{n} \left(m_n^\alpha - \frac{a_j^\alpha}{n} \right) \frac{a_j^{\alpha-2}}{m_n^{3\alpha}} \left[(\alpha-1) m_n^\alpha - \frac{2\alpha}{n} a_j^\alpha \right].\end{aligned}\quad (3.2)$$

Definition 6 (Symmetric Game). *A game is symmetric if the action spaces are identical and the payoff functions are invariant by permutation of clients action and index.*

The next Proposition provides closed-form expression of the Nash equilibrium for some range of parameter α . It also provides the optimal price p_n^* such that no resource is wasted in equilibrium.

Proposition 1. *By direct computation, we have the following results:*

- (i) *The resource sharing game is a symmetric game. All the clients have symmetric strategies in equilibrium whenever it exists.*
- (ii) *For $0 \leq \alpha \leq 1$, the payoff r_j is concave (outside the origin) with respect to own-action a_j . The best response $BR_j(a_{-j})$ is strictly positive and is given by the root of*

$$x^{(\alpha-1)/2} \left(\frac{\alpha c_n}{n p_n} G \right)^{1/2} - \frac{x^\alpha}{n} - G = 0, \quad G \triangleq \frac{1}{n} \sum_{i \neq j} a_i^\alpha$$

where $x \triangleq a_j$ and there is a unique equilibrium (hence a symmetric one) given by $\left(x^{\alpha-1} \frac{\alpha c_n}{n p_n} \frac{n-1}{n} x^\alpha \right)^{\frac{1}{2}} - \frac{x^\alpha}{n} - \frac{n-1}{n} x^\alpha = 0$, i.e.,

$$a_{NE}^* = \alpha \frac{(n-1)c_n}{n^2 p_n}.$$

It follows that the total demand na_{NE}^* at equilibrium is less than $\frac{c_n}{p_n}$ which means that some resources are wasted.

The equilibrium payoff is $\bar{r}_n(a_{NE}^*) = a_j p_n \left[\frac{G + \frac{a_j^\alpha}{n}}{\alpha G} - 1 \right]$ which is positive for $\alpha \leq 1$.

- (iii) For $\alpha > 1$, the activity (participation) of player j depends mainly of the aggregate of the others. $a_j^* > 0$ only if $G \leq G_*$ and the number of active clients should be less than $\frac{\alpha}{\alpha-1}$. If $n > \frac{\alpha}{\alpha-1}$ then $BR_j = 0$.
- (iv) With a participation constraint, the payoff at equilibrium (whenever it exists) is at least 0.
- (v) The game is a symmetric aggregative game in the sense of Dubey et al. 1980, [DMCS80].
- (vi) By choosing the price $p_n^* = \alpha \frac{(n-1)}{n} < \alpha$ one gets that the total demand at equilibrium is exactly the available capacity of the cloud. Thus, pricing design can improve resource sharing efficiency in the cloud. Interestingly, as n goes to infinity, the optimal pricing converges to α .
- (vii) The game is a submodular game with respect to own-action and the mean-field, i.e., it satisfies a decreasing-difference property: if $a_j > a'_j$, then the difference $\bar{r}_n(a_j, m_{n,-j}) - \bar{r}_n(a'_j, m_{n,-j})$ is decreasing with m_n whenever $nm_n^\alpha > a_j^\alpha$.

We say that the cloud renting game is efficient if no resource is wasted, i.e., the equilibrium demand is exactly c_n . Hence, the efficiency ratio is $\frac{na_{NE}^*}{c_n}$. As we can see from (ii) of Proposition 1, the efficiency ratio goes to 1 by setting the price to p_n^* . This type of efficiency loss is due to selfishness and have been widely used in the literature of mechanism design and auction theory. We refer the reader to [NRTV07] for more recent applications.

Note that the equilibrium demand increases with α , decreases with the charged price and increases with the capacity per client. The equilibrium payoff is positive and if $\alpha \leq 1$ each player will participate in an equilibrium.

Note that in case of Nash equilibrium the optimal pricing p_n^* depends on the number of active clients in the cloud and value of α . When the active number of clients varies (for example, due to new entry or exit in the cloud), a new price needs to be setup which is not convenient.

Finite evolutionary stable strategy (F-ESS)

The notion of evolutionary stable strategies (ESS) have been studied in the context of evolutionary games, starting from Volterra 1926 [Vol], Fisher 1931 [FB99], Hamilton 1967 [Ham67] and Maynard Smith & Price in 1973 [MP73]. The original definition was for infinite population in pairwise interaction model. However, the idea can be used for finite population as well. However, the notion of ESS in finite population is slightly different than the ESS notion for infinite population. The reason is that for finite population regime, each generic player has an influence in the mean-field which may result in a non-negligible effect on the relative payoff. We start with a simple (and weaker) notion of Finite evolutionary stable strategy (F-ESS) with one deviant.

Definition 7 (F-ESS). *The pure strategy $a_i^* \forall i \in \mathcal{N}$ is an F-ESS in the cloud game with finite number of players if*

$$r_j(\dots, a_j^*, \dots, a_k, \dots) \geq r_j(\dots, a_j, \dots, a_k^*, \dots)$$

for all $a_i \in \mathcal{A}$, and for all $k \neq j$.

This definition of F-ESS is based on the definition in [Ham67]. This means that if one player $k \neq j$ is adopting a different strategy than the F-ESS i.e. $a_k \neq a_k^*$, the best strategy for user j is to choose the F-ESS $a_j = a_j^*$. The user j does not have interest to deviate from a_j^* . In other words, in a population of $n - 1$ F-ESS players and 1 mutant player, we do not expect the mutant to do better than a typical F-ESS player [Ham67]. Note that this definition of F-ESS does not necessarily imply (Nash) equilibrium condition. For a more detailed description of F-ESS strategies please refer to [Sch88].

Proposition 2. *For $\alpha \leq 1$, the game \mathcal{G}_n has a F-ESS given by $a_{f-ess}^* = \alpha \frac{c_n}{np_n}$.*

Remark 4. *The F-ESS equilibrium a_{f-ess}^* from Proposition 2 is different from the a_{NE}^* from Proposition 1 (ii) which implies that a_{NE}^* is unstable.*

Finite myopic mean-field response (F-MFR)

Here we will provide another strategy which coincides with F-ESS for our reward. Note that this result is not general and is true only for our system model.

Since the cloud game \mathcal{G}_n has an aggregative structure m_n , it is interesting to evaluate the performance of aggregate-taking clients. Each client sees the

aggregate m_n as signal (by ignoring its effect on the signal). A response to the aggregate of such game is called mean-field response. Since the reaction ignores the influence of own-action a_j in m_n , such reaction is called *myopic*. This leads to an equilibrium concept, which we call, finite myopic mean-field response (F-MFR). It corresponds to a mean-field-taking strategy.

Definition 8 (F-MFR). *Formally, a pure F-MFR is a strategy a^* such that*

$$a_j^* \in \arg \max_{a_j \in \mathcal{A}} \left[\frac{c_n}{n} \left(\frac{a_j}{m_n^*} \right)^\alpha - p_n a_j \right],$$

where $m_n^* \triangleq \left(\frac{1}{n} \sum_{i=1}^n (a_i^*)^\alpha \right)^{\frac{1}{\alpha}}$.

This is a fixed-point equation since m_n^* contains the action a_j^* .

Proposition 3. *For $\alpha \leq 1$, the game \mathcal{G}_n has a F-MFR given by*

$$a_{f-mfr}^* = \frac{\alpha c_n}{n p_n}.$$

For $\alpha > 1$ the game has no F-MFR due to inconsistency. Note that a F-MFR is not a Nash equilibrium of \mathcal{G}_n . In the asymptotic regime, a_{f-mfr}^* converges to a mean-field equilibrium.

Remark 5. *The a_{f-mfr}^* and a_{f-ess}^* coincides for $\alpha \leq 1$. However, for $\alpha > 1$ the two notions do not coincide.*

3.3.2 Equilibrium for \mathcal{G}_∞ infinite number of clients

In cloud networks as the number of nodes is very large it is useful to analyze the asymptotic case as it is able to provide simplification and structure to a game which would otherwise be difficult to find. In this section, we characterize the Nash equilibrium for the infinite client which is also referred to as the Wardrop equilibrium [War52], and show that it is a submodular Mean Field (MF) Game. Then we also define an evolutionary stable strategy (ESS) and show that our Nash equilibrium is ESS i.e. it is a stable equilibrium point even for small perturbations in client demand.

In this infinite game, the notion of F-ESS becomes the classical ESS in the sense of Maynard Smith & Price (1973, [MP73]) and the notion of F-MFR becomes a mean-field Nash equilibrium (see Definition 9).

Definition 9 (Mean-field (Nash) equilibrium). m^* is a pure mean-field (Nash) equilibrium if $m^* \in \mathcal{A}$, and m^* is consistent and is a best response to itself: $m^* \in \arg \max_{a_j} \bar{r}(a_j, m^*)$

From the above analysis, we get

Proposition 4. (i) The resource sharing game \mathcal{G}_∞ is a symmetric game.

(ii) the game \mathcal{G}_∞ is a submodular mean-field game, i.e., $\partial_{a_j m}^2 \bar{r} \leq 0$ (at the interior). This means that the incentive to decrease the demand decreases with the level of the m of the mean-field.

(iii) For $\alpha < 1, m > 0$ the best response to mean-field is $br(m) = \left(\frac{\alpha c}{pm^\alpha}\right)^{\frac{1}{1-\alpha}}$ and for $\alpha > 1$ the best response is 0.

(iv) For $\alpha \leq 1$, there is a unique mean-field equilibrium (i.e. global solution) which is given by

$$m^* = \alpha \frac{c}{p},$$

and the optimal pricing is $p^* = \alpha$.

(v) Thus, the Nash equilibrium a_{NE}^* from proposition 1 (ii) can be scaled to converge to the mean-field equilibrium m^* as n grows.

We are now interested in the stability of the mean-field equilibrium. We adapt the notion of evolutionary stable strategy (ESS) introduced by Maynard Smith and Price (1973, [MP73]). The original definition of ESS was for pairwise interaction and bilinear payoffs, here we have a non-linear payoff function (nonlinear in x and non-linear in m).

Definition 10 (ESS). For $\alpha \in [0, 1]$, m^* is an evolutionary stable strategy (ESS), if it cannot be invaded by a fraction of deviants i.e., for any $m \neq m^*$ there exists $1 > \bar{\epsilon}_m > 0$ (which may depend on m) such that for any $\epsilon \in (0, \bar{\epsilon}_m)$ one has

$$\bar{r}(m^*, \epsilon m + (1 - \epsilon)m^*) > \bar{r}(m, \epsilon m + (1 - \epsilon)m^*), \quad m \neq m^*$$

The parameter $\bar{\epsilon}_m$ is called invasion barrier.

For a symmetric game, the reward function $\bar{r}(m^*, \epsilon m + (1 - \epsilon)m^*)$ represents the reward of a user employing the ESS strategy m^* while an ϵ fraction of

players are adopting the strategy of m where $m \neq m^*$ and $1 - \epsilon$ fraction of users are adopting the strategy m^* which makes the effective mean field response to be $\epsilon m + (1 - \epsilon)m^*$ i.e. a convex combination of m and m^* where $m \neq m^*$. Hence the meanfield term in this case is $\epsilon m + (1 - \epsilon)m^*$. The inequality between $\bar{r}(m^*, \epsilon m + (1 - \epsilon)m^*)$ and $\bar{r}(m, \epsilon m + (1 - \epsilon)m^*)$ indicates that the reward of a player implementing m^* is higher as long as ϵ is less than the invasion barrier $\bar{\epsilon}_m$.

Definition 11 (*Invasion barrier $\bar{\epsilon}_m$*). *Invasion barrier $\bar{\epsilon}_m$ is the maximum percentage of the clients which can deviate from m^* such that the system still remains stable i.e. it is able to return to m^* .*

Next, we show that the mean-field equilibrium is an ESS for $\alpha \in (0, 1)$. This means that in the asymptotic regime, the mean-field equilibrium cannot be invaded by a perturbation of deviant VM demands. Moreover, our proof shows that the fraction of deviant(s) can be arbitrary high (i.e. $\epsilon < 1$), which gives a stronger ESS.

Proposition 5. *Let $\alpha \in (0, 1)$. Then, the mean-field equilibrium $m^* = \alpha \frac{c}{p}$ is an evolutionary stable strategy when $\epsilon \neq 1$.*

This is a useful result as it states that as long as the number of deviants is not 100% it is possible for the system to return to the m^* .

3.3.3 Equilibrium for \mathcal{G}_s Demand Satisfaction

Before going for pure satisfactory solution seeking, we first need to ask if the problem is well-posed, i.e; the existence of a pure satisfactory solution i.e. the equilibrium. The feasibility of the satisfaction problem for a given s_j^* , $j \in \mathcal{N}$, is the non-emptiness of the set $\{a = (a_1, \dots, a_n) \in \mathcal{A}^n \mid s_j(a) \geq s_j^*, \forall j\}$. The non-emptiness is equivalent to say that there exists a vector $(\epsilon_1, \dots, \epsilon_n)$, $\epsilon_j \geq 0$ such that there is a demand profile a that satisfies $\forall j, s_j(a) = s_j^* + \epsilon_j$. Thus, a necessary condition for existence of a satisfactory solution is that such a vector $s^* + \epsilon$ belongs to the set $s(\mathcal{A}^n)$, i.e., the range of the vectorial function $s(a) \triangleq (s_1(a), \dots, s_n(a))$. It is not difficult to see that this is also a sufficient condition.

Proposition 6. *Suppose that the satisfaction levels are such that $\sum_{i=1}^n s_i^* < c_n$. Then, the game has a satisfactory solution and the allocation given by*

$$a_i = (s_i^*)^{\frac{1}{\alpha}},$$

satisfies

$$s_j(a) = \frac{c_n s_i^*}{\sum_{j=1}^n s_j^*} > s_i^*$$

The proof is obvious and has been omitted for brevity.

The condition $\sum_{i=1}^n s_i^* < c_n$ translates the fact that the total demand should not exceed the available capacity.

We have proposed a selection algorithm in Section 3.4. The goal of the fully distributed learning algorithm for satisfactory solutions detailed in Subsection 3.4.3, is to select the most efficient satisfactory solution. By most efficient we mean that each user gets exactly the required performance without any waste. Thus, our analysis covers not only the satisfactory solution algorithm but also the problem of selecting the most efficient satisfactory point.

3.4 Algorithms

In the above sections we were able to analyze the equilibria associated with various types of games. As these systems are inherently distributed it is not possible to solve the problem in a centralized fashion. So in this section we provide the iterative distributed algorithms to approach the equilibria described in the above section. These algorithms require limited information in feedback in order to converge to their respective equilibria.

3.4.1 Iterative Learning Algorithm in the Finite Regime

In this subsection we present an iterative learning algorithm which is able to converge to the respective equilibria of the above subsections. The reason for using such an algorithm is that in general, the best response function has a complicated structure (as one can see in section 3.3.1, the equilibrium is a solution of a system of non linear equations.) Therefore, the use of an iterative algorithm allows us to achieve the equilibrium with less computational complexity. In addition the purposed iterative algorithm in this chapter is distributed in the sense that each client updates its own action and sends it to the cloud.

We propose to use an Ishikawa algorithm (see [Ish74] for details) which is represented by the equation as follows

$$a_{t+1} = f(a_t)\lambda + a_t(1 - \lambda) \tag{3.3}$$

where $\lambda > 0$ is a step size $\lambda \in [0, 1]$,

where $f(\cdot)$ is the best response $br(\cdot)$ of the Nash equilibrium. $br(\cdot)$ which is given below:

$$br(a_{j,t-1}, m_{n,t-1}) = \left[\left(\frac{\alpha c_n}{np_n} \left(m_{n,t-1} - \frac{a_{j,t-1}}{n} \right) \right)^{1/2} - \left(m_{n,t-1} - \frac{a_{j,t-1}}{n} \right) \right]_+ \quad (3.4)$$

Note that if a^* is a fixed-point of br i.e. $br(a^*) = a^*$ then $br(a_t)\lambda + a_t(1-\lambda) = a^*\lambda + a^*(1-\lambda) = a^*$. Then a^* is a fixed-point of Equation (3.3).

Proposition 7. *Convergence of the Algorithm to Nash Equilibrium: For a vanishing step size i.e. when $\sum_t \lambda_t = \infty$ and $\sum_t \lambda_t^2 < \infty$ the Ishikawa algorithm (Equation(3.3)) converges to the Nash equilibrium a^* .*

Proof. The proof follows in two steps,

(1) *Convergence to an ODE:* First we show that for vanishing step size, our Ishikawa algorithm (Equation(3.3)) converges to the limiting Ordinary differential equation (ODE) give by

$$\dot{a}_t = br(a_t) - a_t. \quad (3.5)$$

To prove this we can follow the standard ODE technique described in [Tem12].

(2) *Stable point of the ODE:* Second we prove that a^* is a stable point of the ODE. To prove this we just need to verify that the Jacobian J of the R.H.S of the equation (3.5) evaluated at a^* be negative semidefinite i.e. its eigen values have negative real parts. We can show that the diagonal of J is -1 and the non diagonal elements are $\beta = \frac{2-n}{2n(n-1)}$. Hence $\forall y \in \mathbb{R}, y^T J y = -(1+\beta)(\sum_j y_j)^2 - \beta(\sum_j y_j)^2 \leq 0$, for $n > 2$, (where n is the number of players) which implies that J is negative semidefinite which concludes the proof. \square

For more details please refer to [HTAZ12b] where we have presented a similar analysis for in more detail for a similar reward function.

Notice that for a fixed step size (sufficiently small), the Ishikawa algorithm (Equation(3.3)) converges in general for a pseudo contraction mapping [Tem12]. We define the convergence time for a fixed step size (sufficiently small) within a certain η -neighborhood of a^* as the first time the trajectory of the algorithm a_t enters a range within a distance less than η from a^* : $T_\eta = \inf\{t > 0 \mid \|a_t - a^*\|_\infty \leq \eta\}$. The convergence time for η sufficiently small enough, is in order $\log(\frac{1}{\eta})$ [Tem12].

We can apply our Ishikawa based algorithm from equation (3.3) to any of the strategies developed in the previous sections by replacing function $f(\cdot)$ with the respective best response function. This has been omitted due to space limitations.

3.4.2 Iterative Learning Algorithm in the Infinite Regime

Let us develop an algorithm that converges to the stable MF equilibrium by following the same steps as in finite case.

The mean-field response is $a_{j,t+1} \in \arg \max_{a'_j} \bar{r}(a'_j, m_t) = br(m_t)$. In general, this is a multi-valued map (correspondence) as we have seen in the finite number of client case. However, for infinite number of client case (see proposition 5 (iii)), the correspondence is a single-valued map. In this case of a single valued map, equation (3.6) is able to converge to its fixed point. Thus, one can use the following mean-field response learning given by

$$m_{t+1} = f(m_t), \quad m_0 \text{ fixed.} \quad (3.6)$$

We denote $br(\cdot)$ by $f(\cdot)$ to simplify the notation where $br(\cdot)$ represents best response. Note that in Equation (3.6) only the starting point m_0 is required if the player knows the structure of $f(\cdot)$. This means that it is not needed to feedback the mean of the mean-field at each step. Finding a fixed-point of the function $f(\cdot)$ is equivalent to finding the root of $g(m) \triangleq f(m) - m$.

There are several algorithms that may obtain the root of $g(m)$. Each has a different convergence order or time which depends on their construction. We start by defining the convergence order.

Definition 12. Assume that m_t converges to m^* and let $\eta_t \triangleq |\epsilon_t| \triangleq |m_t - m^*|$. If two positive constants $c_1, o > 0$ exist, and $c_1 \triangleq \limsup_{t \rightarrow \infty} \frac{\eta_{t+1}}{\eta_t^o}$,

then the sequence $\{m_t\}_t$ is said to converge to m^* with order of convergence o . The number c_1 is called the asymptotic error constant. The cases $o \in \{1, 2, 3, 4\}$ are given special consideration.

- (i) If $o = 1$ the convergence of $\{m_t\}_t$ is called linear.
- (ii) If $o = 2$ the convergence of $\{m_t\}_t$ is called quadratic.
- (iii) If $o = 3$ the convergence of $\{m_t\}_t$ is called cubic.
- (iv) If $o = 4$ the convergence of $\{m_t\}_t$ is called fourth order.

A classical method that is known to find the root of a single valued map function (e.g. $g(\cdot)$) is the Newton method and it has a convergence order of $o = 2$.

An important requirement for the implementation of the Newton method is the availability of the derivative of the best response which might not be available for some non differentiable functions. Therefore derivative free techniques are used in general.

An example of the derivative free techniques is the so-called *secant method* given by

$$(1.6^*) \begin{cases} m_{t+1} = m_t - \frac{g(m_t)(m_t - m_{t-1})}{g(m_t) - g(m_{t-1})}, \\ m_0 \text{ fixed.} \end{cases}$$

which has the form $m_t = F(m_{t-1}, m_{t-2})$, i.e., a two-step memory scheme. The secant method is known to have a convergence order of $o = 1.6$ (hence, the name of the system (1.6*)). The drawback here is that although this method is derivative free, it is slower than the Newton method in terms of convergence order.

A faster method than the secant method is called the Steffensen method in (2*) which has a second order convergence with only two evaluations per time slot. Interested readers are referred to [DB74] for more details.

$$(2^*) \begin{cases} m_{t+1} = m_t - \frac{g^2(m_t)}{g(m_t + g(m_t)) - g(m_t)}, \\ m_0 \text{ fixed.} \end{cases}$$

In this chapter we go one step further and construct a faster convergence order of $o = 4$. For that we introduce the following notation: $\Delta g[a, a'] \triangleq \frac{g(a) - g(a')}{a - a'}$, $a \neq a'$ and $b_t \triangleq m_t + g(m_t)$. The function $g(\cdot)$ will be evaluated at points: m_t , \bar{y}_t and b_t , where \bar{y}_t is the analogue of the secant point given above. Let $\nu(s) = \alpha_0 + \alpha_1(s - m_t) + \alpha_2(s - m_t)^2$ be a polynomial of degree 2 with coefficients $\alpha_0, \alpha_1, \alpha_2$. We want that $\nu(s)$ approximates the Taylor expansion of function $g(\cdot)$ till order 2. To achieve that, it is sufficient to determine the coefficients $\alpha_0, \alpha_1, \alpha_2$ using the following requirement: $\nu(m_t) \triangleq g(m_t)$, $\nu(\bar{y}_t) = g(\bar{y}_t)$ and $\nu(b_t) = g(b_t)$ (for more details one can refer to [DB74]). From these relations, we deduce the coefficients

- $\alpha_0 = g(m_t)$
- $\alpha_1 = \frac{\bar{y}_t \Delta g[b_t, m_t] + m_t \Delta g[\bar{y}_t, m_t] - m_t \Delta g[b_t, m_t] - b_t \Delta g[\bar{y}_t, m_t]}{\bar{y}_t - b_t}$
- $\alpha_2 = \frac{\Delta g[b_t, m_t] - \Delta g[\bar{y}_t, m_t]}{b_t - \bar{y}_t}$

Note that the derivative of ν at the point m_t is exactly the coefficient α_1 . In other words, α_1 approximates the derivative of $g(\cdot)$ at point m_t (since ν is a Taylor approximation of $g(\cdot)$). Now, if we consider the Newton update method and replace the derivative of $g(\cdot)$ by coefficient α_1 (α_1 can be seen as a plausible derivative-free term of $g(\cdot)$), we obtain our derivative-free update algorithm given as follows,

$$(4^*) \left\{ \begin{array}{l} m_{t+1} = \bar{y}_t - \frac{(1+w_t)(\bar{y}_t - b_t)g(\bar{y}_t)}{((\bar{y}_t - m_t)\Delta g[b_t, m_t] - (b_t - m_t)\Delta g[\bar{y}_t, m_t])} \\ \\ \text{where } \bar{y}_t \text{ is the secant point} \\ \bar{y}_t \triangleq m_t - \frac{g(m_t)}{\Delta g[b_t, m_t]}, \\ w_t \triangleq 2 \frac{g(\bar{y}_t)}{g(b_t)} \end{array} \right.$$

Properties 1. *The proposed speedup scheme (4^*)*

- *is a derivative-free scheme,*
- *has three evaluations of the function $g(\cdot)$ per time-slot.*
- *is memoryless*
- *is faster than the secant speedup method (superlinear order),*
- *is faster than Steffensen's speedup learning method (quadratic order).*

Proposition 8. *Assume that $g(\cdot)$ has a simple zero in \mathcal{A} . Locally, the scheme (4^*) has a convergence order equal to four.*

Definition 13 (Convergence time). *The convergence time to be within an η -neighborhood of m^* is the first time the trajectory enters into this neighborhood, i.e.,*

$$T_{\eta, \epsilon_0} \triangleq \inf\{t > 0 \mid |m_t - m^*| \leq \eta, |m_0 - m^*| = \epsilon_0\}$$

Proposition 9. *The convergence time is $T_{\eta, \epsilon_0} = 1 + \lfloor \max(0, T) \rfloor$ where*

$$T = \frac{1}{\ln(4)} \ln \left[\frac{\ln \left(\frac{1}{\eta c_1^{\frac{1}{3}}} \right)}{\ln \left(\frac{1}{\epsilon_0 c_1^{\frac{1}{3}}} \right)} \right]$$

which is in order of $O \left(\log(\log(\frac{1}{\eta})) \right)$.

Where c_1 is a positive constant as represented in definition 12 and whose value is given in the proof of Proposition 8. *The result of Proposition 9 is very important since strictly contraction mapping fixed-point algorithm exhibits a convergence time in order of $O(\log(\frac{1}{\eta}))$. Notice that the convergence time of Ishikawa algorithm in finite case is $\log(\frac{1}{\eta})$ [Tem12]. Here we have a bound in order of $\log(\log(\frac{1}{\eta}))$ which is a great improvement.*

3.4.3 Iterative Learning Algorithm for satisfactory solution

We propose a fully distributed learning algorithm for satisfactory solutions. The only information assumption required to each client is the numerical realized value of its satisfaction $s_{j,t}$ and its own-satisfaction level s_j^* .

We propose to use the following fixed-point iteration given by

$$a_{j,t+1} = \text{proj}_{\mathcal{A}} \left[a_{j,t} \frac{s_j^*}{s_{j,t}} \right], \quad (3.7)$$

where $\text{proj}_{\mathcal{A}}(x) = \min(c_n, \max(0, x))$. $\text{proj}_{\mathcal{A}}$ is the projection operator over the nonempty and convex set \mathcal{A} . The above is Banach Picard algorithm and (see [Wil70] for details) and it converges to the equilibrium characterized in Proposition 6.

sketch of proof for convergence: . The algorithm (3.7) is *fully distributed* in the sense that a client does not need to observe the actions of the others in order to update its strategy iteratively. The righthand side of (3.7) is denoted by f_j . The function f_j is well-defined whenever the ratio $\frac{a_j}{s_j} s_j^* = \frac{s_j^*}{c_n} a_j^{1-\alpha} (\sum_{i=1}^n a_i^\alpha)$ has a limit when $a_j \rightarrow 0$ if $\alpha \leq 1$. Due to space limit, we omit the proof that the function in (3.7) is a fixed point equation (for more details see [DB74]). The set of fixed-point of f is denoted by $\text{fix}(f)$ is contained in the set of satisfactory solution. It is clear that if the algorithm converges to some interior point a^* then a^* is a satisfactory solution. To prove this statement, consider a converging sequence to a^* . Combining the continuity of the projection map and the continuity of the payoff function, one gets the continuity of $\text{proj}_{\mathcal{A}} \left[a_j \frac{s_j^*}{s_j} \right]$. Taking the limit as t goes to infinity yields $a_j = \text{proj}_{\mathcal{A}} \left[a_j \frac{s_j^*}{s_j} \right]$, i.e., the payoff of client j is $s_j = s_j(a^*) = s_j^*$ which means that every satisfied client j is in interior steady state. \square

For the infinite case as $n \rightarrow +\infty$ with satisfactory level s^* , the algorithm simplifies to $m_{t+1} = \frac{s^*}{c} m_t + \frac{s^*}{c} \epsilon m_t^{1-\alpha}$ where $\epsilon > 0$ is very small, which converges to an interior fixed-point.

3.5 Numerical Investigation

For the numerical investigation we fix $\alpha = 1$.

3.5.1 Finite user scenario

In the finite case we numerically show that Nash equilibrium is unstable in the sense that it is different from the ESS equilibrium. For that, we compare the performance of the Nash equilibrium and the ESS equilibrium. Figure 3.1 compares the evolution of the demand a_j (top) and the reward r_j (bottom) for $c_n = 1000, n = 10$ clients with the same price $p_n = 11$. The horizontal dotted red line shows the Nash equilibrium and the horizontal solid red line represents the ESS equilibrium. The dotted and solid blue lines correspondingly converge to their horizontal red lines, which shows that Nash equilibrium is different from ESS and hence Nash equilibrium is not a stable point in the finite case.

We then examine the case when different clients are using different strategies. Figure 3.2 shows the scenario where 1 client is using the ESS strategy and $n - 1$ clients are using the Nash strategy (where the number of clients $n = 10$). This figure shows that ESS dominates Nash in the sense that ESS client represented by the solid green line gets a higher reward than the other $n - 1$ clients and also has a higher demand, while the case when all clients are implementing Nash strategy are represented by the solid blue lines for reference. It is clear from the plots that the performance of users in interactive Nash equilibrium strategy which is represented by dotted blue line as indicated in the legend is even lower than the solid blue line which represents the pure Nash when all users are using the same Nash strategy. It should be noted that as we have the same price for all users we only see 3 lines as all the other users have the same demand and the same reward which would result in the same equilibrium.

3.5.2 Infinite user scenario

Next we numerically illustrate that in the infinite case i.e. for large number of clients Nash and ESS equilibria converge to the same point. Here we extend the case of from Figure 3.2 to $n = 1000$ and present in Figure 3.3 the evolution of the client demand a_j (top) and client reward r_j (bottom). Note that the capacity $c_n = 100000$ and the price is $p_n = 11$ for all clients. It can be clearly seen that the trajectories of Nash and ESS equilibria converge to the same point which

means that Nash equilibrium is stable for large number of users i.e. the mean field case. It should be noted that as we have the same price we only plot one user of each type for clarity.

3.5.3 Heterogeneous pricing scenario

In this section we numerically extend the analysis to the case of different pricing (i.e. heterogeneous pricing scenario) and show that our framework works with different price scenarios. We have included some numerical results to corroborate our claim. Notice that these results are obtained by considering a modified reward function where a different price per user is used, then we find the best-response function for this case and we feed these new best responses in the Ishikawa based algorithm.

In Figure 3.5 we presents the plot for the evolution of the demand a_j (top) and the reward r_j (bottom) for different prices in the finite case with $n = 4$. The prices are $p_1 = 19.4684$, $p_2 = 10.5606$, $p_3 = 18.3646$ and $p_4 = 10.2081$. which are generated randomly. For clarity we only plot the first two users with prices $p_1 = 19.4684$ and $p_2 = 10.5606$ (represented by color blue and green respectively in the legend) to demonstrate our point. As the users have different prices it leads to different Nash equilibria represented in the plots by (o) markers when all users are employing Nash equilibrium strategy. Then we also illustrate the curves when all users are employing ESS strategy represented in the plots by (\square) marker. Here we numerically demonstrate two points at once. Firstly, our framework is applicable to the finite heterogeneous case i.e. for different price scenario. Secondly the ESS and Nash are different in the heterogeneous finite scenario.

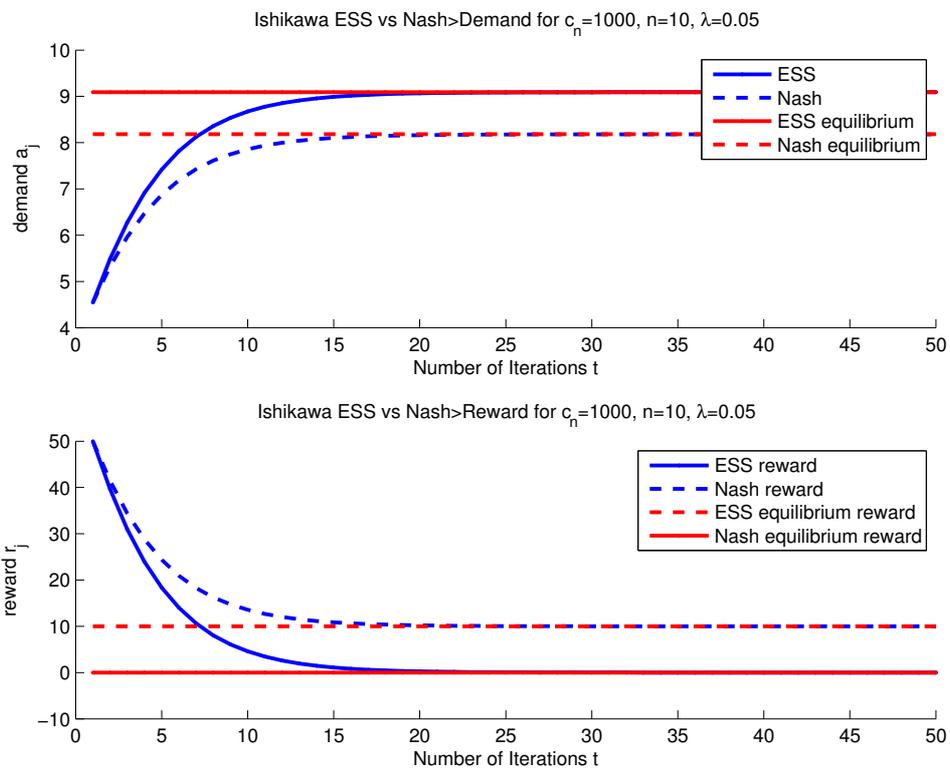


Figure 3.1: ESS vs Nash $n = 10$

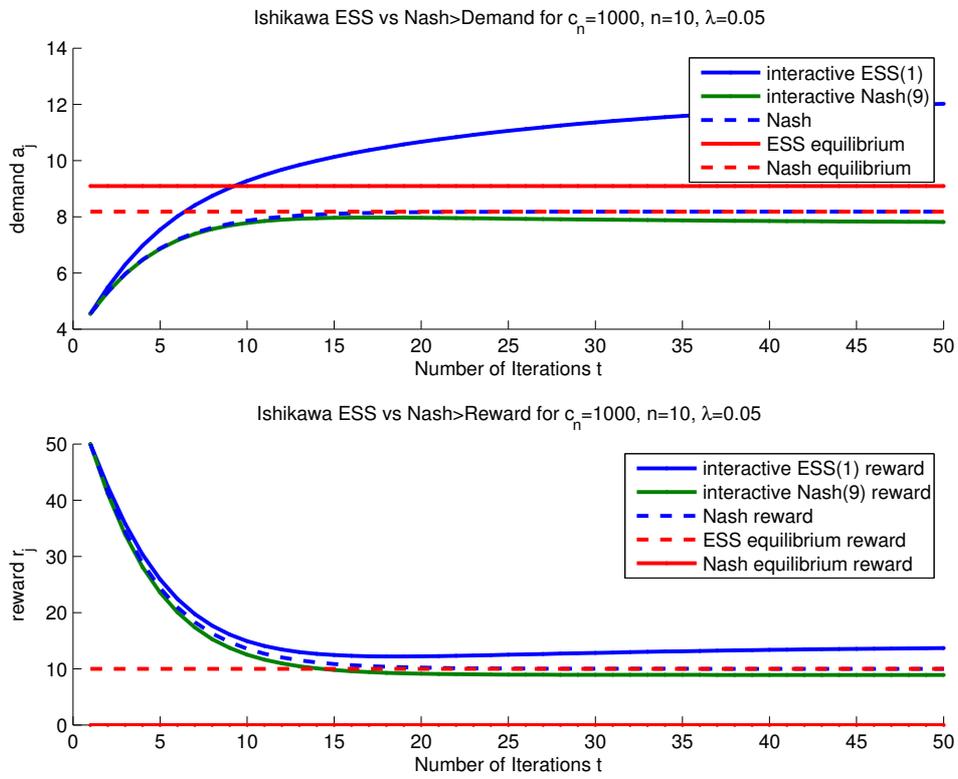


Figure 3.2: ESS vs Nash $n = 10$ Interactive

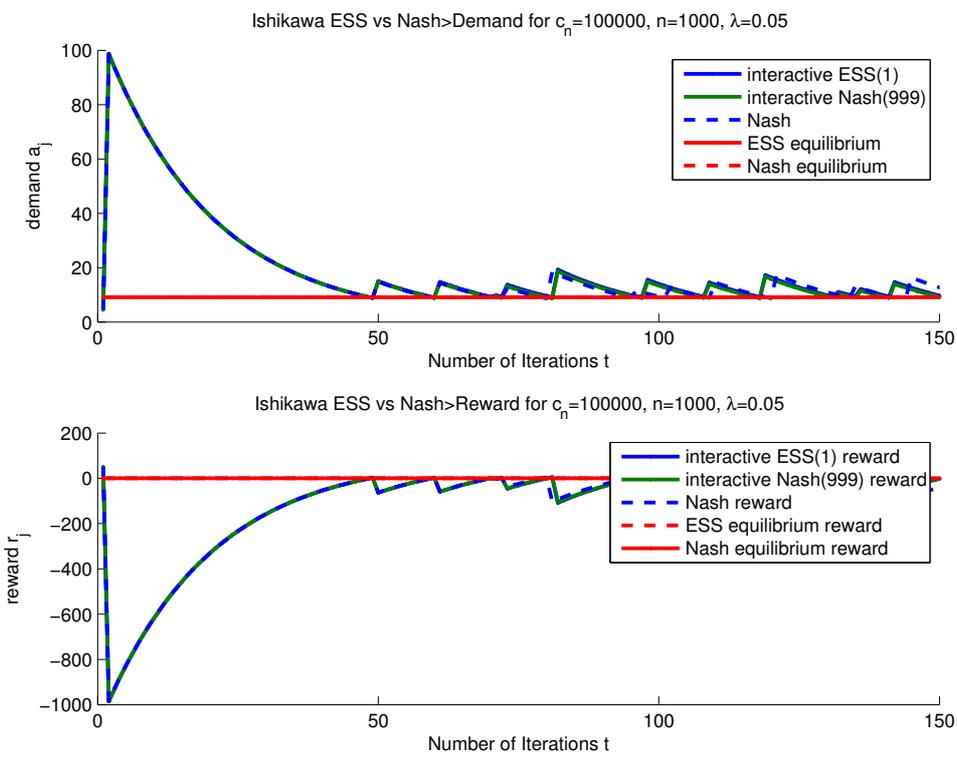


Figure 3.3: ESS vs Nash $n = 1000$

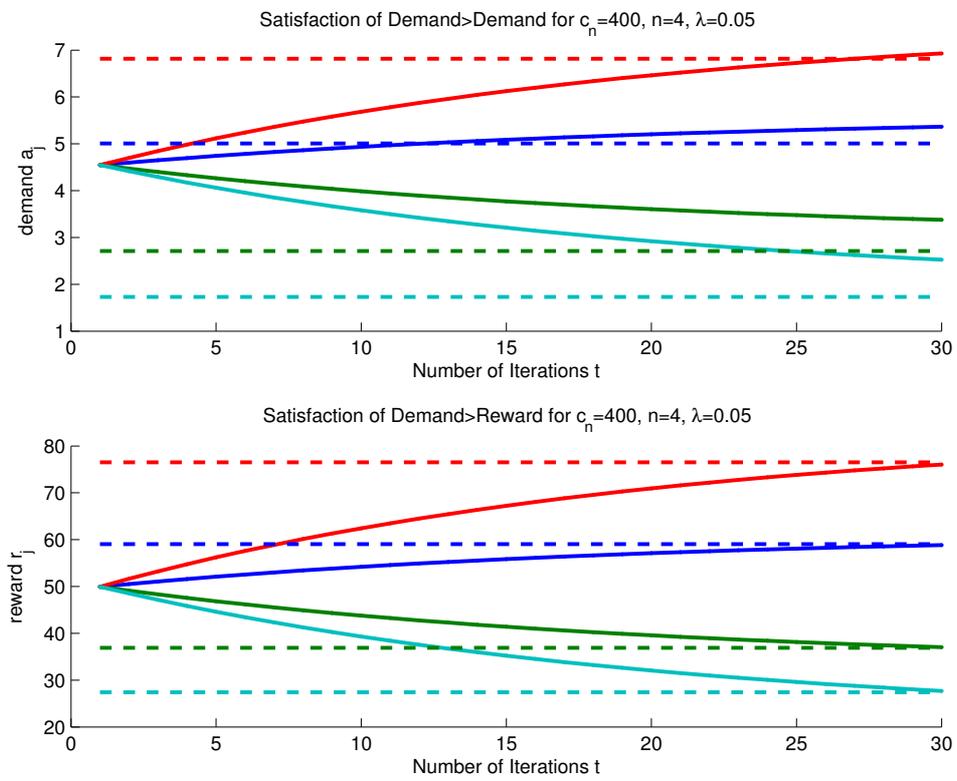


Figure 3.4: Satisfaction $n = 4$

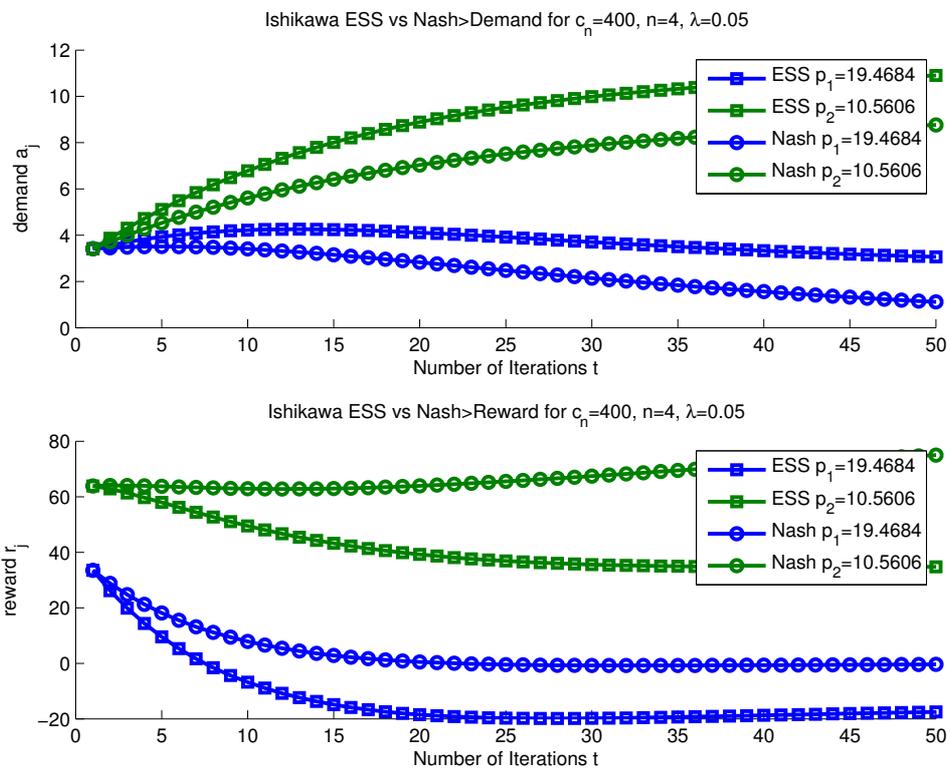


Figure 3.5: Ishikawa Iteration for Different Price i.e. heterogeneous case

Chapter 4

Conclusion and Future Work

Future networks will have to handle higher data rate, offer lower latency and efficient spectrum and bandwidth utilization in an energy efficient manner. This will require a paradigm shift in the fundamental architecture of the network, which will make individual nodes more independent, intelligent, and able to make decisions about how, when and where to communicate. This distributed network architecture poses several design challenges. In this thesis we have tried to address a subset of those challenges such that it could be possible to make intelligent decisions based on the limited information and still able to converge to a local maximum equilibrium point.

4.1 Extremum Seeking

In this section we summarize our *extremum seeking* framework with contributions and their implications. We consider a distributed systems scenario with stochastic state dependent reward function, which needs to satisfy Lipschitz continuity condition. Each node has a numerical value of the reward available in feedback at each time slot. We have extended the classical Nash Seeking with sinus perturbation method [FKB12b] to the case of discrete time and stochastic state-dependent payoff functions. Our contribution is the development of a deterministic perturbation algorithm which is able to converge locally to a state independent Nash equilibrium according to Theorem 1 in Chapter 2 for vanishing step size and provide an error bound according to Theorem 2 in Chapter 2 for fixed step size. We also provide the convergence time for the *extremum seeking* algorithm according to Corollary 1 in Chapter 2. The error bound for

the algorithm and the convergence time to be in a close neighborhood of the Nash equilibrium..

Algorithm: For ‘model free’ scenarios (i.e. when the model of the reward function is not known by all the nodes) we have developed a stable *extremum seeking* algorithm that is able to converge to the Nash equilibrium when initialized in the neighborhood of the Nash equilibrium point. Our proposed *discrete time* learning algorithm uses sinus perturbation, for *continuous* action games where each node has only a numerical realization of own-payoff at each time. This *extremum seeking* algorithm is able to find the local minima using just the numerical value of the stochastic state dependent payoff function at each discrete time sample. Note that since the payoff function may not necessarily be concave, finding a global optimum in reasonable time can be difficult in general even in deterministic case (fixed state) and known closed-form expression of payoff. The algorithm ensures tractability which can be a requirement in certain engineering applications.

Following is a list of the three distributed wireless system scenarios that we have considered. Numerical results for each of these cases was also provided for illustration.

- Distributed power control in femto cells using bayesian density tracking.
- Distributed transmit beamforming with 1-bit feedback for LoS-MISO channels.
- Distributed stochastic learning for continuous power control in wireless networks.

For certain applications where the reward function may not satisfy the Lipschitz continuity condition, we may still be able to use our *extremum seeking* algorithm, but may not mathematically be able to prove or ensure the convergence or stability. This may not be acceptable in some mission critical applications, and may be acceptable in others.

4.2 Resource Sharing Games in Cloud Networks

In this section we summarize our Resource Sharing Framework with contributions and their implications.

Resource Sharing Problem: For resource sharing in cloud networks with finite number of user, each users has a specific reward which needs to be maximized. For our specific type of resource sharing reward function, we have proved that Nash equilibrium point is unstable for our reward function and have identified two stable strategies namely finite evolutionary stable strategy (F-ESS) and finite myopic mean field response (F-MFR). We extend this to the asymptotic case, i.e. when the number of users approaches infinity, then the Nash equilibrium point becomes evolutionary stable, which means that it cannot be dominated or invaded by another competing strategy.

Satisfaction Problem: We consider another type of resource utilization problem where each users has a minimum demand which need to be satisfied. These type of models, more accurately capture the requirements of the users, where the maximum requirement of the users is known. We provide necessary and sufficient conditions for the existence of satisfactory solution where each user is satisfied anywhere anytime.

Algorithms and Numerical Results: We provide an algorithms which are able to respectively converge to the equilibria mention above. We develop a faster iterative distributed algorithm that converges to the mean-field equilibrium with a convergence order of four. We also develop a distributed iterative algorithm for large scale systems using Mean Field theory and apply them to finite scale systems. In large scale systems for the same price we were able to show that the reward for a social users in better than reward for a selfish user.

The resource sharing problem is a general problem that arises in various application domains. We are able to derive fundamental results which scale to large scale systems. We have studied resource sharing games in cloud networks, however our analysis is applicable to other resources sharing scenarios as well. The reward function considered here has a very specific structure which is based on the percentage of resources available to each node. There may be scenarios where the percentage of resources can be re-interpreted as the access probability for random access scenarios.

4.3 Future Work

The future work could focus on

- the extension of the *extremum seeking* framework to non-Lipschitz continu-

ous reward functions, as Lipschitz continuity is not a very easy condition to satisfy in most engineering applications. This extension if possible would enable our framework to be applicable to a wider range of engineering problems.

- Additionally the *extremum seeking* framework could be extended to the case of vectors of action, where each users is able to perform multiple actions based on multiple rewards. This type of extension could be very useful in systems where there are multiple inputs to a system.
- So far all of the results achieved are for the local stability, it would be interesting to extend the analysis to global stability condition for Nash equilibrium for both deterministic and stochastic payoff functions.
- the resource sharing framework in cloud networks could be extended to dynamic pricing where the pricing function depends on the user's demand. This type of dynamic pricing reward would be more flexible in dealing with dynamic demand of the users and will be able to incentivise the uniform distribution of the demand thus discouraging peak load outages.

Appendix A

Proof of convergence in Extremum Seeking

A.1 Variable Step Size: Proof of Theorem 1

The Theorem 1 states that Under Assumption A1, the learning algorithm converges almost surely to the trajectory of a non-autonomous system given by

$$\begin{aligned}\frac{d}{dt}\hat{a}_{j,t} &= z_j b_j \sin(\Omega_j t + \phi_j) \mathbb{E}_{\mathbf{S}} (r_j(\mathbf{S}, \mathbf{a}_t)) \\ a_{j,t} &= \hat{a}_{j,t} + b_j \sin(\Omega_j t + \phi_j)\end{aligned}$$

The proof follows in several steps.

- The first step provides conditions for Lipschitz continuity of the expected payoff which is given in Lemma 1. From Lemma 2 we have that $\forall j, t$, $f_j(t, \mathbf{a}) := b_j z_j \sin(\Omega_j t + \phi_j) \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$, is Lipschitz over the domain \mathcal{A}
- Second step: the learning rates are chosen such that they satisfy assumption A1.
- Third step: we check the noise conditions.

Lemma 1. *Let*

$$\begin{aligned}(\mathbf{S}, \mathbf{a}) &\longmapsto r_j(\mathbf{S}, \mathbf{a}) \quad \forall \mathbf{S} \in \mathcal{S}, \exists L_{j,\mathbf{S}} \text{ such that} \\ (C_1) &: \|r_j(\mathbf{S}, \mathbf{a}) - r_j(\mathbf{S}, \mathbf{a}')\| \leq L_{j,\mathbf{S}} \|\mathbf{a} - \mathbf{a}'\| \quad \forall (\mathbf{a}, \mathbf{a}') \in \mathcal{A}, \\ &\quad \mathbf{a} \neq \mathbf{a}' \\ (C_2) &: \mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}} < +\infty\end{aligned}$$

then the mapping $\mathbf{a} \mapsto \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is Lipschitz with Lipschitz constant $L_j := \mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}}$

Proof of Lemma 1. Suppose that $\mathbf{a} \mapsto E_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is Lipschitz with Lipschitz constant $L_{j,\mathbf{S}}$, then by Jensen's inequality one has

$$\|\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}')\| \leq \mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{a}) - r_j(\mathbf{S}, \mathbf{a}')\|$$

By condition C_2 , $\mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}} < +\infty$. Let L_j be $\mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}}$. Then

$$\|\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}')\| \leq L_j \|\mathbf{a} - \mathbf{a}'\|$$

This completes the proof.

Remark 6. • Note that under C_1 and C_2 the expected payoff vector $\mathbf{r} := (r_j)_{j \in \mathcal{N}}$ is Lipschitz continuous with $L := \max_j L_j$,

- If \mathcal{S} is a compact set and $\mathbf{S} \mapsto L_{j,\mathbf{S}}$ is continuous then $\mathbf{a} \mapsto \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is Lipschitz [In particular, the condition C_2 is not needed]

We shall prove the above remark by *Reductio ad absurdum*. To prove the second statement of Remark 6 we use compactness and continuity argument. We start from Bolzano–Weierstrass theorem which states that: For any k , any continuous map $\mathbf{a} \mapsto \mathbf{f}(k, \mathbf{a})$ over a compact set \mathcal{A} has at least one maximum, i.e., $\sup \mathbf{f}(k, \mathbf{a}) = \max_{\mathbf{a} \in \mathcal{A}} \mathbf{f}(k, \mathbf{a}) < \infty$. The proof of this statement can be easily done by contradiction. Suppose $\sup \mathbf{f}(k, \mathbf{a}) = \infty$. Then there exists a sequence $(\mathbf{a}_l)_l$ such that $\mathbf{a}_l \in \mathcal{A}$ but $\mathbf{f}(k, \mathbf{a}_l) \rightarrow \infty$ as l goes to infinity. This is impossible because \mathcal{A} is compact which implies that $\mathbf{f}(k, \mathcal{A}) = \{\mathbf{f}(k, \mathbf{a}) \mid \mathbf{a} \in \mathcal{A}\}$ is bounded by continuity.

Since \mathcal{S} is compact and $\mathbf{S} \mapsto L_{\mathbf{S}}$ is continuous, $\sup_{\mathbf{S} \in \mathcal{S}} L_{\mathbf{S}}$ is also finite.

Remark 7. If $r_j(\mathbf{S}, \mathbf{a})$ is continuously differentiable with the respect to \mathbf{a} then it is sufficient to check that the expectation of the gradient of $r_j(\mathbf{S}, \mathbf{a})$ is bounded (in norm).

if \mathcal{S} is in Euclidean Space

- $r_j(\mathbf{S}, \mathbf{a})$ is differentiable w.r.t \mathbf{a}
- $r_j(\mathbf{S}, \mathbf{a}), \nabla_{\mathbf{a}} r_j(\mathbf{S}, \mathbf{a})$ are continuous in \mathbf{S}

- $r_j(\mathbf{S}, \mathbf{a}), \nabla_{\mathbf{a}} r_j(\mathbf{S}, \mathbf{a})$ are absolutely integrable in \mathbf{S} and $\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a})$ is continuous in \mathbf{a} .

then

$$\mathbb{E}[\nabla_{\mathbf{a}} r_j(\mathbf{S}, \mathbf{a})] = \nabla_{\mathbf{a}} \mathbb{E}[r_j(\mathbf{S}, \mathbf{a})]$$

which can be written as

$$\int_{\mathbf{S}} \nabla_{\mathbf{a}} r_j(\mathbf{S}, \mathbf{a}) \gamma(d\mathbf{S}) = \nabla_{\mathbf{a}} \int_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}) \gamma(d\mathbf{S})$$

where γ is the measure of \mathbf{S} state space. For more details on the above conditions please refer to [SR64].

Since $f_j(\cdot)$ is a function of time and the actions of nodes, we need a uniform Lipschitz condition on $f_j(\cdot)$.

We have

$$|f_j(t, \mathbf{a}) - f_j(t, \mathbf{a}')| \leq b_j z_j |\sin(\Omega_j t + \phi)| [|\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}')|]$$

But one has $|\sin(\cdot)| \leq 1$. Hence,

$$|f_j(t, \mathbf{a}) - f_j(t, \mathbf{a}')| \leq b_j z_j [|\mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}) - \mathbb{E}_{\mathbf{S}} r_j(\mathbf{S}, \mathbf{a}')|]$$

by using Lemma 1 we get,

$$|f_j(t, \mathbf{a}) - f_j(t, \mathbf{a}')| \leq b_j z_j L_j \|\mathbf{a} - \mathbf{a}'\|$$

This implies that the Lipschitz constant of f_j is less than the one of r_j times the factor $b_j z_j$.

Finally, we check the noise conditions. The recursion equation is given by

$$a_{j,k+1} = a_{j,k} + \lambda_k [f_j(k, \mathbf{a}_k) + M_{j,k+1}]$$

where $M_{j,k+1}$ is a martingale difference sequence. By definition the martingale sequence for the algorithm is given as

$$M_{j,k+1} := z_j b_j \sin(\Omega_j t_k + \phi_j) [\tilde{r}_{j,k+1} - \mathbb{E}_{\mathbf{S}} [\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})]]$$

which satisfied the condition $\mathbb{E}[M_{k+1} | \mathcal{F}_k] = 0$ for $k \geq 0$ almost surely (a.s.)

Lemma 2. If $\mathbf{a}_k \in \mathcal{A}$ then the martingale is square-integrable with

$$\mathbb{E}[\|M_{k+1}\|^2 | \mathcal{F}_k] \leq \acute{c}(1 + \|\mathbf{a}_k\|^2) \forall k$$

Proof of Lemma 2. Let $\tilde{r}_{j,k+1}$ be the realization the payoff at time $k + 1$. The expected value of this random variable can be bounded above the norm of \mathbf{a}_k .

Let $M_{j,k+1} := z_j b_j \sin(\Omega_j t_k + \phi_j)(\tilde{r}_{j,k+1} - \mathbb{E}_{\mathbf{S}}[\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})])$

$$\begin{aligned} \|M_{j,k+1}\| &\leq |z_j| |b_j| |(\sin(\Omega_j t_k + \phi_j))| \|\tilde{r}_{j,k+1} \\ &\quad - \mathbb{E}_{\mathbf{S}}[\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})]\| \\ &\leq z_j b_j (\|\tilde{r}_{j,k+1}\| + \|\mathbb{E}_{\mathbf{S}}[\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})]\|) \\ &\leq z_j b_j (\|\tilde{r}_{j,k+1}\| + \mathbb{E}_{\mathbf{S}}\|\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})\|) \\ &\leq z b (\|\tilde{r}_{j,k+1}\| + \mathbb{E}_{\mathbf{S}}\|\tilde{r}_{j,k+1}(\mathbf{S}, \mathbf{a}_{k+1})\|) \end{aligned}$$

Where $|\sin(\cdot)| \leq 1$, $z := \max |z_j|$, $b := \max |b_j|$, $\|\tilde{r}_{j,k+1}\|$ is bounded because of the Lipschitz condition as mentioned in C_1 , which is shown below.

$$\begin{aligned} \|r_j(\mathbf{S}, \mathbf{a}_k) - r_j(\mathbf{S}, \mathbf{0})\| &\leq L_{j,\mathbf{S}} \|\mathbf{a}_k - \mathbf{0}\| \quad \forall (\mathbf{a}_k) \in \mathcal{A} & \text{(A.1)} \\ \|r_j(\mathbf{S}, \mathbf{a}_k)\| &\leq \|r_j(\mathbf{S}, \mathbf{0})\| + L_{j,\mathbf{S}} \|\mathbf{a}_k\| \\ &\leq \beta_{1,\mathbf{S}} + L_{j,\mathbf{S}} \|\mathbf{a}_k\| \end{aligned}$$

Where $\beta_{1,\mathbf{S}} := \|r_j(\mathbf{S}, \mathbf{0})\|$. The above equations A.1 show that $\|r_j(\mathbf{S}, \mathbf{a})\|$ is bounded by $\beta_{1,\mathbf{S}} + L_{j,\mathbf{S}} \|\mathbf{a}\|$. By taking expectation of the above set of inequalities we get.

$$\begin{aligned} \mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{a}_k)\| &\leq \mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{0})\| + \mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}} \|\mathbf{a}_k\| & \text{(A.2)} \\ &\leq L_j \|\mathbf{a}_k\| + \mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{0})\| \\ &\leq L_j \|\mathbf{a}_k\| + \beta_2 \end{aligned}$$

Where $\beta_2 := \mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{0})\|$, $L_j := \mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}}$. The above set of inequalities A.2 show that $\mathbb{E}_{\mathbf{S}} \|r_j(\mathbf{S}, \mathbf{a})\|$ is bounded.

Combining the results of inequalities in (A.1) (A.2) we can get

$$\begin{aligned} \|M_{j,k+1}\|^2 &\leq z^2 b^2 (\beta_{1,\mathbf{S}} + L_{j,\mathbf{S}} \|\mathbf{a}_k\| + L_j \|\mathbf{a}_k\| + \beta_2)^2 \\ &\leq 2z^2 b^2 ((\beta_{1,\mathbf{S}} + \beta_2)^2 + (L_{j,\mathbf{S}} + L_j)^2 \|\mathbf{a}_k\|^2) \\ &\leq 4z^2 b^2 (\beta_{1,\mathbf{S}}^2 + \beta_2^2 + (L_{j,\mathbf{S}}^2 + L_j^2) \|\mathbf{a}_k\|^2) \end{aligned}$$

Taking $\mathbb{E}_{\mathbf{S}}$ over the above inequalities we get:

$$\begin{aligned} \mathbb{E}_{\mathbf{S}} \|M_{j,k+1}\|^2 &\leq 4z^2 b^2 (\mathbb{E}_{\mathbf{S}} \beta_{1,\mathbf{S}}^2 + \beta_2^2 + (\mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}}^2 + L_j^2) \|\mathbf{a}_k\|^2) \\ &\leq 4z^2 b^2 (\beta + \acute{L}_j \|\mathbf{a}_k\|^2) \\ &\leq \acute{c} (1 + \|\mathbf{a}_k\|^2) \end{aligned}$$

Where $\acute{L}_j := \mathbb{E}_{\mathbf{S}} L_{j,\mathbf{S}}^2 + L_j^2$, $\beta := \mathbb{E}_{\mathbf{S}} \beta_{1,\mathbf{S}}^2 + \beta_2^2$ and $4z^2 b^2 (\beta + \acute{L}_j) \leq \acute{c}$

This completes the proof. \square

We now combine the above three steps to derive almost sure convergence to an ODE. To do so, we interpolate the stochastic process \mathbf{a}_k (an affine interpolation) in order to get a continuous time process following the lines of Borkar [Bor08] Chapter 2 Lemma 1 for the case of non-autonomous system. The solution of the non-autonomous differential equation

$$\frac{d}{dt} \mathbf{a}_t = \mathbf{f}(t, \mathbf{a}_t)$$

is \mathbf{a}_t^* . The gap between the \mathbf{a}_t^* and interpolated process is given by

$$\lim_{t_k \rightarrow \infty} \sup_{t \in [t_k, t_k + T]} \|\bar{\mathbf{a}}_t - \mathbf{a}_t^*\| = 0 \text{ a.s.}$$

i.e. it vanishes almost surely (a.s.) for asymptotic interval of length $T > 0$ In order to calculate the bound we need to define a few terms which are helpful in obtaining a compact form of the bound.

$$\begin{aligned} \sup_{t \in [t_k, t_k + T]} \|\bar{\mathbf{a}}_t - \mathbf{a}_t^{t_k}\| &\leq K_{T,k} e^{LT} + C_T \sup_{\acute{k} \geq 0} \lambda_{k+\acute{k}} \\ &= C_T (\lambda_{k+\acute{k}} + L \sum_{\acute{k} \geq 0} \lambda_{k+\acute{k}}^2) \\ &\quad + \sup_{\acute{k} \geq 0} \|\delta_{k,k+\acute{k}}\| \end{aligned}$$

where

$$\begin{aligned}
 K_{T,k} &:= C_T L \sum_{\acute{k} \geq 0} \lambda_{k+\acute{k}}^2 + \sup_{\acute{k} \geq 0} \|\delta_{k,k+\acute{k}}\| \\
 \delta_{k,k+\acute{k}} &:= \xi_{k+\acute{k}} - \xi_k \\
 \xi_k &:= \sum_{\acute{k}=0}^{k-1} \lambda_{\acute{k}} M_{\acute{k}+1} \\
 C_T &:= \|\mathbf{r}_0\| + L(C_0 + \|\mathbf{r}_0\|T)e^{LT} < \infty \\
 L &:= \max_{j \in \mathcal{N}} \mathbb{E}_{\mathbf{S}}[L_j, \mathbf{S}] \\
 \mathbf{r}_k &:= [r_{1,k}, \dots, r_{N,k}] \\
 \mathbb{P} \left(\sup_{\acute{k}} \|\mathbf{a}_{\acute{k}}\| < C_0 \right) &= 1
 \end{aligned}$$

The above bound is derived following the steps of the proof of Lemma 1 in chapter 2 of [Bor08]. From the above bound we can conclude that

$$\lim_{t_k \rightarrow \infty} \sup_{t \in [t_k, t_k+T]} \|\bar{\mathbf{a}}_t - \mathbf{a}_t^*\| = 0 \text{ a.s.}$$

A.2 Fixed Step Size: Proof of Theorem 2

Theorem 2 states that under Assumption A2, the learning algorithm converges in distribution to the trajectory of a non-autonomous system given by

$$\begin{aligned}
 \frac{d}{dt} \hat{a}_{j,t} &= z_j b_j \sin(\Omega_j t + \phi_j) \mathbb{E}_{\mathbf{S}}(r_j(\mathbf{S}, \mathbf{a}_t)) \\
 a_{j,t} &= \hat{a}_{j,t} + b_j \sin(\Omega_j t + \phi_j)
 \end{aligned}$$

Proposition 1. *Let $\bar{\mathbf{a}}_t$ be the interpolated version the trajectory of our algorithm at time t and $\hat{\mathbf{a}}_t$ is the trajectory of the the ODE at time t . Under assumption A2 $\bar{\mathbf{a}}_t$ converges to $\hat{\mathbf{a}}_t$ as step size λ vanishes i.e. $\lambda \rightarrow 0$.*

$$\mathbb{E} \sup_{t \in [0, T]} [\|\bar{\mathbf{a}}_t - \hat{\mathbf{a}}_t\|^2]^{\frac{1}{2}} = \tilde{C}_T \sqrt{\lambda}$$

Proposition 1 implies Theorem 2.

Proof of Proposition 1. To prove the above proposition we start with a fixed step size $\lambda > 0$.

- Time Scale. $\tilde{t}_k := \sum_{\acute{k}=1}^k \lambda = k\lambda$, for $k \geq 0$

- The cumulative noise at iteration k is $\xi_k = \sum_{\acute{k}=0}^{k-1} \lambda M_{\acute{k}+1} = \lambda \sum_{\acute{k}=0}^{k-1} M_{\acute{k}+1}$
- Define the (affine) interpolated process from $\{\hat{\mathbf{a}}_{\acute{k}}\}_{\acute{k} \geq 0}$ rewritten as

$$\hat{a}_{j,k+1} = \hat{a}_{j,k} + \lambda(f_j(k, \hat{\mathbf{a}}_k) + M_{k+1}).$$

The advantage of the interpolated process is that it is defined for any continuous time by concatenation. The affine interpolation writes $\bar{\hat{\mathbf{a}}}_t := \hat{\mathbf{a}}_k + (\frac{t-\tilde{t}_k}{\lambda})(\hat{\mathbf{a}}_{k+1} - \hat{\mathbf{a}}_k)$ if $t \in [k\lambda, (k+1)\lambda[$ which is now in continuous time.

Note that constant learning rate or constant step size $\lambda_k = \lambda$ is suitable for many practical scenarios. It is used for example in numerical analysis: Euler's Scheme (1st Order), Runge Kutta's scheme (4th Order), etc. Our algorithm writes

$$(**) \begin{cases} \hat{a}_{j,k+1} &= \hat{a}_{j,k} + \lambda b_j z_j \sin(\tilde{t}_k \Omega_j + \phi_j) \tilde{r}_{j,k+1} \\ a_{j,k} &= \hat{a}_{j,k} + b_j \sin(\tilde{t}_k \Omega_j + \phi_j) \end{cases}$$

where λ is a constant learning rate, our aim is to analyze (**) asymptotically when λ is very small. In order to prove an asymptotic pseudo-trajectory result for constant learning rate, we need additional assumptions for the sequence generated by the powers. The key additional assumption is the uniform integrability of that process. We need the conditions C_1 C_2 , which translate into

- From Remark 7: gradient of the expectation of payoff is bounded
- From Lemma 2: Square of the martingale is bounded
- Uniform Integrability of $r_j(\mathbf{S}, \mathbf{a})$

and $\hat{\mathbf{a}}_t$ is the solution of $\dot{\hat{\mathbf{a}}}_t = \mathbf{f}(t, \hat{\mathbf{a}}_t)$ starting from $\hat{\mathbf{a}}_{\lfloor \frac{t}{\lambda} \rfloor}$

$$\begin{aligned} \bar{\hat{\mathbf{a}}}_{\tilde{t}_{k+T}} &= \sum_{\acute{k}=1}^T \underbrace{(\bar{\hat{\mathbf{a}}}_{\tilde{t}_{k+\acute{k}}} - \bar{\hat{\mathbf{a}}}_{\tilde{t}_{k+\acute{k}-1}})}_{= \lambda(\mathbf{f}(\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor}) + M_{\acute{k}+1})} \\ &\quad + \bar{\hat{\mathbf{a}}}_{\tilde{t}_k} \end{aligned}$$

Using the recursive stochastic equation $(\bar{\hat{\mathbf{a}}}_{\tilde{t}_{k+\acute{k}}} - \bar{\hat{\mathbf{a}}}_{\tilde{t}_{k+\acute{k}-1}}) = \lambda(\mathbf{f}(\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor}) + M_{k+\acute{k}+1})$ one gets

$$\begin{aligned}
 \tilde{\mathbf{a}}_{\tilde{t}_{k+T}} &= \sum_{\acute{k}=1}^T (\tilde{t}_{k+\acute{k}} - \tilde{t}_{k+\acute{k}-1}) \mathbf{f}(\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor}) \\
 &\quad + \sum_{\acute{k}=1}^T \lambda M_{k+\acute{k}+1} + \tilde{\mathbf{a}}_{\tilde{t}_k},
 \end{aligned} \tag{A.3}$$

By rewriting $(\tilde{t}_{k+\acute{k}} - \tilde{t}_{k+\acute{k}-1}) = \int_{\tilde{t}_{k+\acute{k}-1}}^{\tilde{t}_{k+\acute{k}}} ds$ the last expression can be expanded as

$$\begin{aligned}
 \tilde{\mathbf{a}}_{\tilde{t}_{k+T}} &= \sum_{\acute{k}=1}^T \int_{\tilde{t}_{k+\acute{k}-1}}^{\tilde{t}_{k+\acute{k}}} \mathbf{f}(\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor}) ds \\
 &\quad + \sum_{\acute{k}=1}^T \lambda M_{k+\acute{k}+1} + \tilde{\mathbf{a}}_{\tilde{t}_k}
 \end{aligned}$$

This implies that

$$\begin{aligned}
 \tilde{\mathbf{a}}_{\tilde{t}_{k+T}} &= \sum_{\acute{k}=1}^T \int_{\tilde{t}_{k+\acute{k}-1}}^{\tilde{t}_{k+\acute{k}}} \mathbf{f}(\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{\tilde{t}_{k+\acute{k}-1}}{\lambda} \rfloor}) ds \\
 &\quad + (\xi_{k+T} - \xi_k) + \tilde{\mathbf{a}}_{\tilde{t}_k} \\
 \tilde{\mathbf{a}}_{\tilde{t}_{k+T}} &= \sum_{\acute{k}=1}^T \int_{\tilde{t}_{k+\acute{k}-1}}^{\tilde{t}_{k+\acute{k}}} \mathbf{f}(\lfloor \frac{s}{\lambda} \rfloor, \hat{\mathbf{a}}_{\lfloor \frac{s}{\lambda} \rfloor}) ds \\
 &\quad + (\xi_{k+T} - \xi_k) + \tilde{\mathbf{a}}_{\tilde{t}_k}
 \end{aligned}$$

Now we use Burkholder's inequality which states the following: For an $\alpha > 0$ there exists two constants $c_1 > 0$ and $c_2 > 0$ such that

$$\begin{aligned}
 c_1 \mathbb{E}[\sum_{\acute{k}=1}^k \|\hat{\mathbf{a}}_{\acute{k}} - \hat{\mathbf{a}}_{\acute{k}-1}\|^2]^{\alpha/2} &\leq \mathbb{E}[\sup_{\acute{k} \leq k} \|\hat{\mathbf{a}}_{\acute{k}}\|]^\alpha \\
 &\leq c_2 \mathbb{E}[\sum_{\acute{k}=1}^k \|\hat{\mathbf{a}}_{\acute{k}} - \hat{\mathbf{a}}_{\acute{k}-1}\|^2]^{\alpha/2}
 \end{aligned}$$

A direct application to the process $\eta_k := \lambda \sum_{\acute{k} \leq k} \|M_{\acute{k}}\|^2$ gives

$$\begin{aligned}
 c_1 \mathbb{E}[\sum_{\acute{k}=1}^k \|\eta_{\acute{k}} - \eta_{\acute{k}-1}\|^2]^{\alpha/2} &\leq \mathbb{E}[\sup_{\acute{k} \leq k} \|\eta_{\acute{k}}\|]^\alpha \\
 &\leq c_2 \mathbb{E}[\sum_{\acute{k}=1}^k \|\eta_{\acute{k}} - \eta_{\acute{k}-1}\|^2]^{\alpha/2}
 \end{aligned}$$

We consider the inequality of η_k and we use discrete Gronwall inequality which states that if

$$\epsilon_{k+1} \leq C + L \sum_{\acute{k}=0}^k \lambda \epsilon_{\acute{k}}$$

where $C, L, \epsilon_k > 0 \forall k \geq 0$ then one gets

$$\epsilon_{k+1} \leq C e^{L\lambda k}$$

Taking

$$\epsilon_k := \mathbb{E}[\sup_{\acute{k} \leq k} \|\bar{\hat{\mathbf{a}}}_{t_{\acute{k}}} - \hat{\mathbf{a}}_{t_{\acute{k}}}\|^2]^{1/2}$$

in the above inequality and setting

$$C := \lambda T K_1 \sqrt{1 + C_0^2} + \sqrt{\lambda K_2 (1 + C_0^2)},$$

$$L := \max_{j \in \mathcal{N}} \mathbb{E}_{\mathbf{s}}[L_j, \mathbf{s}]$$

for some $K_1, K_2 = c_2$ with

$$K_1 := \max(c_1, c_2 \sqrt{1 + C_0^2}), C_0 = \mathbb{E}[\sup_{\acute{k} \geq 0} \|\bar{\hat{\mathbf{a}}}_{\acute{k}}\|^2]^{1/2}$$

yields to

$$\mathbb{E}[\sup_{\acute{k} \leq k} \|\bar{\hat{\mathbf{a}}}_{t_{\acute{k}}} - \hat{\mathbf{a}}_{t_{\acute{k}}}\|^2]^{1/2} \leq \sqrt{\lambda} \tilde{C}_T$$

This shows that $\mathbb{E}[\sup_{\acute{k} \leq k} \|\bar{\hat{\mathbf{a}}}_{t_{\acute{k}}} - \hat{\mathbf{a}}_{t_{\acute{k}}}\|^2]^{1/2}$ is bounded and implies Proposition 1. When $\lambda \rightarrow 0$ we have a weak convergence of the interpolated process to a solution of the ODE. The error gap is $\sqrt{\lambda} \tilde{C}_T$ which vanishes as $\lambda \rightarrow 0$.

This completes the proof. \square

A.3 Stability analysis, Proof of Theorem 3

The proof of theorem 3 follows the steps in [FKB12b]. The main steps are provided here.

We denote the error relative to the Nash equilibrium as $\tilde{a}_{j,t} = \hat{a}_{j,t} - \mu_j(t) - a_j^*$ and formulate the error system in the time scale $\tau = \underline{\Omega}t$.

$$\frac{d}{d\tau} \tilde{a}_{j,\tau} = z_j \mu_j(\tau) \bar{f}_j(\tilde{a}_{j,\tau} + \mu(\tau) + a^*) \quad (\text{A.4})$$

Applying averaging theory to equation (A.4) we can write it as follows, where $\tilde{a}_{j,\tau}^{avg}$ represents the average action of node j .

$$\frac{d}{d\tau} \tilde{a}_{j,\tau}^{avg} = \epsilon Z_j \lim_{T \rightarrow 0} \frac{1}{T} \int_0^T z_j \mu_j(\tau) \bar{f}_j(\tilde{a}^{avg} + \mu(\tau) + a^*) d\tau \quad (\text{A.5})$$

The equilibrium $\tilde{a}^e = [\tilde{a}_1^e, \dots, \tilde{a}_N^e]$ of (A.5) satisfies

$$0 = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T z_j \mu_j(\tau) \bar{f}_j(\tilde{a}^e + \mu(\tau) + a^*) d\tau \quad (\text{A.6})$$

for all $j \in \{1, \dots, N\}$ we postulate that \tilde{a}_j^e (which is the equilibrium of the average system) has the form

$$\tilde{a}_j^e = \sum_{m=1}^N \alpha_j^m b_j + \sum_{m=1}^N \sum_{t \geq m} \beta_j^{mn} b_m b_t + O(\max_j b_j^3) \quad (\text{A.7})$$

By expanding \bar{f}_j about \mathbf{a}^* in (A.6) and substituting (A.7) the unknown coefficients α_j^m and β_j^{mn} can be determined.

The Taylor series expansion of \bar{f}_j about \mathbf{a}^* in (A.6) for an N-player game is

$$\bar{f}_j(a^* + \nu) = \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \frac{\nu_1^{t_1} \dots \nu_N^{t_N}}{t_1! \dots t_N!} \left(\frac{\partial^{t_1 + \dots + t_N} \bar{f}_j}{\partial a_1^{t_1} \dots \partial a_N^{t_N}} \right) (a^*) \quad (\text{A.8})$$

Where $\nu_j = \tilde{a}_j^e + \mu_j(\tau)$

substituting (A.8) in (A.6) and computing the average of each term gives

$$0 = \frac{1}{T} \int_0^T z_j \mu_j(\tau) \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \frac{\nu_1^{t_1} \dots \nu_N^{t_N}}{t_1! \dots t_N!} \left(\frac{\partial^{t_1 + \dots + t_N} \bar{f}_j}{\partial a_1^{t_1} \dots \partial a_N^{t_N}} \right) (a^*) d\tau \quad (\text{A.9})$$

$$= \frac{1}{T} \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \int_0^T z_j \mu_j(\tau) \frac{\nu_1^{t_1}, \dots, \nu_N^{t_N}}{t_1! \dots t_N!} \left(\frac{\partial^{t_1 + \dots + t_N} \bar{f}_j}{\partial a_1^{t_1} \dots \partial a_N^{t_N}} \right) (a^*) d\tau \quad (\text{A.10})$$

$$= \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \int_0^T \frac{\mu_j(\tau) (\mu_1(\tau) + \tilde{a}_1^e)^{t_1}, \dots, (\mu_N(\tau) + \tilde{a}_N^e)^{t_N}}{t_1! \dots t_N!} \underbrace{\left(\frac{\partial^{t_1 + \dots + t_N} \bar{f}_j}{\partial a_1^{t_1} \dots \partial a_N^{t_N}} \right)}_{\phi(t_1, \dots, t_N)} (a^*) d\tau \quad (\text{A.11})$$

$$= \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \int_0^T \frac{\mu_j(\tau) (\mu_1(\tau) + \tilde{a}_1^e)^{t_1}, \dots, (\mu_N(\tau) + \tilde{a}_N^e)^{t_N}}{t_1! \dots t_N!} \phi(t_1, \dots, t_N) d\tau \quad (\text{A.12})$$

$$= \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \underbrace{\frac{\phi(t_1, \dots, t_N)}{t_1! \dots t_N!}}_{\Phi_{(t_1, \dots, t_N)}} \int_0^T \mu_j(\tau) (\mu_1(\tau) + \tilde{a}_1^e)^{t_1}, \dots, (\mu_N(\tau) + \tilde{a}_N^e)^{t_N} d\tau \quad (\text{A.13})$$

$$= \sum_{t_1=1}^{\infty} \dots \sum_{t_N=1}^{\infty} \Phi_{t_1, \dots, t_N} \int_0^T \mu_j(\tau) (\mu_1(\tau) + \tilde{a}_1^e)^{t_1}, \dots, (\mu_N(\tau) + \tilde{a}_N^e)^{t_N} d\tau \quad (\text{A.14})$$

we are able to exploit the orthogonality of the perturbation frequencies. Now we expand the terms up to third order.

Before computing the derivatives of these terms we need to present some simple results, which are based on the conditions $\Omega_j \neq \Omega_m$, $\Omega_j \neq 2\Omega_m$, $\Omega_j \neq \Omega_m + \Omega_t$

$$\frac{1}{T} \int_0^T \mu_j d\tau = 0 \quad (\text{A.15})$$

$$\frac{1}{T} \int_0^T \mu_j^2 d\tau = \frac{b_j^2}{2} \quad (\text{A.16})$$

$$\frac{1}{T} \int_0^T \mu_j^3 d\tau = 0 \quad (\text{A.17})$$

$$\frac{1}{T} \int_0^T \mu_j^4 d\tau = \frac{3b_j^4}{8} \quad (\text{A.18})$$

$$\frac{1}{T} \int_0^T \mu_T \mu_j d\tau = 0 \quad (\text{A.19})$$

$$\frac{1}{T} \int_0^T \mu_T \mu_t \mu_j d\tau = 0 \quad (\text{A.20})$$

$$\frac{1}{T} \int_0^T \mu_T \mu_j^2 d\tau = 0 \quad (\text{A.21})$$

$$\frac{1}{T} \int_0^T \mu_T^2 \mu_j^2 d\tau = \frac{b_m^2 b_j^2}{4} \quad (\text{A.22})$$

$$\frac{1}{T} \int_0^T \mu_T \mu_t \mu_j d\tau = 0 \quad (\text{A.23})$$

By Simplification we get.

$$\begin{aligned} 0 &= \frac{b_j^2}{2} \left[\tilde{a}_j^e \frac{\partial^2 \bar{f}_j}{\partial a_j^2}(a^*) + \sum_{m \neq j}^N \tilde{a}_j^e \frac{\partial^2 \bar{f}_j}{\partial a_j \partial a_m}(a^*) \right. \\ &\quad + \left(\frac{1}{2} (\tilde{a}_j^e)^2 + \frac{b_j^2}{8} \right) \frac{\partial^3 \bar{f}_j}{\partial a_j^3}(a^*) + \tilde{a}_j^e \sum_{m \neq j}^N \tilde{a}_T^e \frac{\partial^3 \bar{f}_j}{\partial a_j^2 \partial a_m}(a^*) \\ &\quad + \sum_{m \neq j}^N \left(\frac{1}{2} (\tilde{a}_j^e)^2 + \frac{b_j^2}{4} \right) \frac{\partial^3 \bar{f}_j}{\partial a_j \partial a_m^2}(a^*) \\ &\quad \left. + \sum_{m \neq j}^N \sum_{\substack{t > m \\ t \neq j}}^N \tilde{a}_T^e \tilde{a}_t^e \frac{\partial^3 \bar{f}_j}{\partial a_j \partial a_m \partial a_t}(a^*) \right] + O(\max_j b_j^5) \quad (\text{A.24}) \end{aligned}$$

Here we have noted assumption A3, put in the value of (A.7) and computed the average of each term.

Substituting (A.6) in (A.26) and matching the first order powers of b_j gives

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = b_1 \Lambda \begin{bmatrix} \alpha_1^1 \\ \vdots \\ \alpha_1^N \end{bmatrix} + \dots + b_N \Lambda \begin{bmatrix} \alpha_1^1 \\ \vdots \\ \alpha_N^N \end{bmatrix} \quad (\text{A.25})$$

which implies that $\alpha_m^j = 0$ for all j, m such since Λ is nonsingular by Assumption 4. Similarly matching the second order terms of b_j and substituting $\alpha_m^j = 0$ we get

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \sum_{m=1}^N \sum_{t>m}^N b_m b_t \Lambda \begin{bmatrix} \beta_{mn}^1 \\ \vdots \\ \beta_{mn}^N \end{bmatrix} + \sum_{m=1}^N b_m^2 \left(\Lambda \begin{bmatrix} \beta_{mm}^1 \\ \vdots \\ \beta_{mm}^N \end{bmatrix} + \pi_m \right) \quad (\text{A.26})$$

$$[\beta_{mm}^1, \dots, \beta_{mm}^{T-1}, \beta_{mm}^T, \beta_{mm}^{m+1}, \dots, \beta_{mm}^N]^T = -\Lambda^{-1} \pi_m \quad (\text{A.27})$$

$$\pi_m = \left[\frac{1}{4} \frac{\partial^3 \bar{f}_1}{\partial a_1 \partial a_m^2}(a^*), \dots, \frac{1}{4} \frac{\partial^3 \bar{f}_{T-1}}{\partial a_{T-1} \partial a_m^2}(a^*), \frac{1}{8} \frac{\partial^3 \bar{f}_T}{\partial a_m^3}(a^*), \frac{1}{4} \frac{\partial^3 \bar{f}_{m+1}}{\partial a_{m+1} \partial a_m^2}(a^*), \dots, \frac{1}{4} \frac{\partial^3 \bar{f}_N}{\partial a_N \partial a_m^2}(a^*) \right]^T \quad (\text{A.28})$$

thus $\beta_{mn}^j = 0$ for all j, m, n when $m \neq n$ and β_{mm}^j is given by (A.27). The equilibrium of the average system is

$$\boxed{\tilde{a}_j^e = \sum_{m=1}^N \beta_j^{mm} b_m^2 + O(\max_j b_j^3)} \quad (\text{A.29})$$

By again utilizing Taylor series expansion one can show that the Jacobian $\Psi^{ave} = [\psi_{j,m}]_{N \times N}$ of (A.5) at \tilde{a}_j^e has elements given by

$$\begin{aligned} \psi_{j,m} &= \epsilon Z_j \lim_{T \rightarrow 0} \frac{1}{T} \int_0^T \mu_j(\tau) \frac{\partial \bar{f}_j}{\partial a_m} (\tilde{a}^e + \mu(\tau) + a^*) d\tau \\ &= \frac{1}{2} \epsilon Z_j \frac{\partial^2 \bar{f}_j}{\partial a_j \partial a_m} (a^*) + O(\epsilon \max_j b_j^3) \end{aligned} \quad (\text{A.30})$$

which is Hurwitz by assumption A3 and A4 for sufficiently small b_j , which means that the equation (A.29) of the average system (A.5) is locally exponentially stable, i.e. there exist $M, m > 0$ such that

$$|\tilde{a}^{ave}(\tau) - \tilde{a}^e| < Me^{-tm}|\tilde{a}^{ave}(0) - \tilde{a}^e|.$$

From general averaging theory (Khalil [2002], Fink [1947]) the system (A.5) retains the stability properties of the average system.

This completes the proof.

A.4 Conditions for Example A

Following are some details about how to obtain \mathbf{a}^* for our application.

$$g_{ij} := |h_{ij}|^2$$

$$\bar{g}_{ij} := \mathbb{E}_{\mathbf{g}} g_{ij} = \mathbb{E}_{\mathbf{g}} |h_{ij}|^2$$

From remark 7 we can write $\mathbb{E}_{\mathbf{G}} \frac{\partial \gamma_j(\mathbf{G}, \mathbf{a}^*)}{\partial a_j} = \frac{\partial}{\partial a_j} \mathbb{E}_{\mathbf{G}} \gamma_j(\mathbf{G}, \mathbf{a}^*) = 0$. Solving N equations we have the following matrix form.

$$\mathbf{a}^* := \begin{pmatrix} \mathbf{a}_1^* \\ \mathbf{a}_2^* \\ \vdots \\ \mathbf{a}_N^* \end{pmatrix}, \bar{\mathbf{G}} := \begin{pmatrix} \bar{g}_{11} & \bar{g}_{12} & \cdots & \bar{g}_{1N} \\ \bar{g}_{21} & \bar{g}_{22} & \cdots & \bar{g}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{g}_{N1} & \bar{g}_{N2} & \cdots & \bar{g}_{NN} \end{pmatrix},$$

$$\mathbf{b} := \begin{pmatrix} \frac{\omega \bar{g}_{11}}{\lambda} - \sigma^2 \\ \frac{\omega \bar{g}_{22}}{\lambda} - \sigma^2 \\ \vdots \\ \frac{\omega \bar{g}_{NN}}{\lambda} - \sigma^2 \end{pmatrix}$$

The above equation can be written in the compact form as

$$\mathbf{a}^* = \bar{\mathbf{G}}^{-1} \mathbf{b}$$

$\bar{\mathbf{G}}$ should be invertible and all the elements in the vector $\bar{\mathbf{a}}$ should be strictly positive as they are a linear combination of power and gains which are positive. We can also write $\omega \bar{g}_{jj} > \lambda \sigma^2$. For this example we can write

$$\mathbb{E}_{\mathbf{G}} [g_{jj}] > \sum_{j' \neq j} \mathbb{E}_{\mathbf{G}} [g_{jj'}] \quad \forall j, j' \neq j$$

If this condition is satisfied then $\bar{\mathbf{G}}$ is invertible.

As G is a matrix of random channel gains it is almost surely invertible. To show the invertibility of this matrix we just need to show that the $\det(\bar{\mathbf{G}}) \neq 0$. The set of invertible matrices \mathcal{G} where $G \in \mathcal{G}$.

Appendix B

Proofs in Mean Field Resource Sharing

Proof of Proposition 1: The proof is in several steps:

(i) The payoff function r_j of the cloud game (given in Equation (3.1)) is obtained by permutation of index of the players and the actions. Moreover, the action space is common. Hence, the game is symmetric.

There is no asymmetric interior equilibrium. Since the game is reduced to a subgame between the active players only, all the players will use symmetric strategies in equilibrium.

(ii) A direct computation of the first order and second order derivatives of r_j shows that $\alpha \in [0, 1]$ is a sufficient condition for the concavity of the payoff with respect to own-action, i.e., $x_j \rightarrow r_j(x_j, x_{-j})$ is concave. With the participation constraint, we need to examine only the possible interior equilibria. Since it is an open set, the derivative should vanish. This means that

$$x^{(\alpha-1)/2} \left(\frac{\alpha c_n}{n p_n} G \right)^{1/2} - \frac{x^\alpha}{n} - G = 0, \quad G \triangleq \frac{1}{n} \sum_{i \neq j} a_i^\alpha$$

and the unique symmetric interior equilibrium candidate is $a_{NE}^* = \alpha \frac{n-1}{n} \frac{c_n}{p_n}$. Note that *the equilibrium demand increases with α , decreases with the charged price and increases with the capacity per user*. The equilibrium payoff is positive and if $\alpha \leq 1$ each player will participate in equilibrium.

(iii) Using the equilibrium payoff at the equilibrium candidate from (ii), one gets that $G + \frac{a_j^\alpha}{n} > \alpha G$ which means that the aggregative term from the other players should not exceed a certain quantity G_* and $\alpha < 1 + \frac{1}{n-1}$, It trivially limits the number of users to be $n < \frac{\alpha}{\alpha-1}$.

(iv) Nonparticipation into the game will result to zero-payoff. Thus, the equilibrium payoff is positive or zero by dominance.

(v) The game is an aggregative game. A simple of aggregative games is the one in which the payoff function can be written into the following form : $\tilde{r}_j(a_j, \phi(a))$, where ϕ is the aggregative term. Our game belongs to that class with the aggregative term being m_n .

(vi) The total demand in equilibrium is $\alpha \frac{(n-1)c_n}{np_n}$. Thus, by designing the price p_n to be $p_n^* = \alpha \frac{n-1}{n}$ the equilibrium demand is to total available resource c_n , i.e., no resource is wasted in equilibrium. The optimal price $p_n^* \rightarrow \alpha$ as n grows.

(vii) A direct differentiation at the interior of the domain shows that $\partial_{a_j \tilde{G}}^2 r_j \leq 0$ whenever $G > a_j^\alpha$ where $\tilde{G} = \sum_i a_i^\alpha$. This completes the proof of Proposition 1. \square

Proof of Proposition 2: The relative payoff is

$$c_n \frac{a_j^\alpha - (a^*)^\alpha}{\frac{a_j^\alpha}{n} + (n-1)(a^*)^\alpha} - p_n(a_j - a^*)$$

By differentiating with the respect to a_j , a first order condition equivalent $a_{j-ess}^* = \alpha \frac{c_n}{np_n}$.

The second order condition is satisfied for $\alpha \leq 1$. \square

Proof of Proposition 3: Simple differentiation of the payoff function (first and second order) yields to $\frac{\alpha c_n}{np_n} a^{\alpha-1} \frac{1}{m_n^\alpha} - 1 = 0$ and the second derivative is negative for $\alpha \leq 1$. If $\alpha > 1$ the mean-field taking strategy leads to unboundedness or non-participation. \square

Proof of Proposition 4: (i) The first item statement can be proved following similar lines as above.

(ii) the cross derivative $\partial_{a_j m}^2 \bar{r}$ is given by $-\alpha^2 c a_j^{\alpha-1} m^{-\alpha-1} \leq 0$. Hence the game is submodular.

(iii) the best response to mean-field is obtained by direct computation.

(iv) The fixed-point is $m^* = \frac{\alpha c}{p}$.

(v) the equilibrium converges to $m^* = \frac{\alpha c}{p}$ whenever $\frac{c_n}{n} \rightarrow c$ and $p_n \rightarrow p$ as n goes to infinity.

This completes the proof of Proposition 4.

Proof of Proposition 5: Let $m \neq m^*$ and $m_\epsilon \triangleq \epsilon m + (1-\epsilon)m^* = m^* + \epsilon(m - m^*)$ for $\epsilon \neq 1$. It follows that $m^* - m = \frac{m^* - m_\epsilon}{\epsilon}$. Let A be the quantity $\bar{r}(m^*, m_\epsilon) - \bar{r}(m, m_\epsilon)$. Then,

$$A = p \left[\frac{c}{p} \left(\frac{(m^*)^\alpha - m^\alpha}{m_\epsilon^\alpha} \right) - (m^* - m) \right] \quad (\text{B.1})$$

$$= \frac{p}{\alpha m_\epsilon^\alpha} [m^*((m^*)^\alpha - m^\alpha) - \alpha(m^* - m)m_\epsilon^\alpha] \quad (\text{B.2})$$

where we have used that the fact that $\frac{c}{p} = \frac{(m^*)}{\alpha}$. The function $x \mapsto x^\alpha$ is concave in \mathcal{A} . Therefore, $(m^*)^\alpha - m^\alpha > \alpha(m^*)^{\alpha-1}(m^* - m)$. By multiplying by m^* both sides, one gets $m^*((m^*)^\alpha - m^\alpha) > \alpha(m^*)^\alpha(m^* - m)$. Thus,

$$A > \frac{p}{\alpha m_\epsilon^\alpha} [\alpha(m^*)^\alpha(m^* - m) - \alpha(m^* - m)m_\epsilon^\alpha] \quad (\text{B.3})$$

$$= \frac{p}{m_\epsilon^\alpha} (m^* - m) [(m^*)^\alpha - m_\epsilon^\alpha] \quad (\text{B.4})$$

$$= \frac{p}{m_\epsilon^\alpha} \frac{m^* - m_\epsilon}{\epsilon} [(m^*)^\alpha - m_\epsilon^\alpha] \quad (\text{B.5})$$

$$= \frac{p}{\epsilon m_\epsilon^\alpha} (m^* - m_\epsilon) [(m^*)^\alpha - m_\epsilon^\alpha] > 0 \quad (\text{B.6})$$

where the last inequality follows because x^α is nondecreasing for $\alpha > 0$ and $m^* \neq m_\epsilon$. We conclude that $\bar{r}(m^*, \epsilon m + (1-\epsilon)m^*) > \bar{r}(m, \epsilon m + (1-\epsilon)m^*)$ for any $\epsilon \in (0, 1)$ and $m \neq m^*$. In particular, the result is true for the case of interest i.e., for $\epsilon \in (0, \bar{\epsilon}_m)$. This completes the proof. \square

Proof. Proof of Proposition 8: If the function $g(\cdot)$ is sufficiently many times differentiable with simple zero, and m_0 is a initial guess that is close enough to m^* ,

Let us define the error $\epsilon_{t+1} \triangleq m_{t+1} - m^*$ and $q_j = \frac{\mathfrak{d}^{(j)}g(m^*)}{j!}$ where $\mathfrak{d}^{(j)}$ is a j th order differential operator.

We claim that ϵ_{t+1} can be expressed as

$$\epsilon_{t+1} = c_2 \epsilon_t^4 + O(\epsilon_t^5)$$

where $c_2 \triangleq \frac{(1+q_1)q_2[(5+3q_1)q_2^2 - q_1(1+q_1)q_3]}{q_1^3}$

This means that locally, the scheme (4*) has fourth-order convergence rate (see Definition 12). Note that using the notation from Definition 12, we have $c_1 \triangleq |c_2|$ and $\eta_{t+1} = |\epsilon_{t+1}|$.

Lets now prove the above claim:

Using equation (4*) we can write equation (B.7)

$$\epsilon_{t+1} = \epsilon_t - \frac{g(m_t)}{A_1} - (1 + w_t) \frac{g(\bar{y}_t)}{A_2} \quad (\text{B.7})$$

where A_1 and A_2 are respectively given by equation (B.8) (B.9)

$$A_1 \triangleq \Delta g[b_t, m_t] \quad (\text{B.8})$$

$$A_2 \triangleq \frac{(\bar{y}_t - m_t) \Delta g[b_t, m_t] - (b_t - m_t) \Delta g[\bar{y}_t, m_t]}{\bar{y}_t - b_t} \quad (\text{B.9})$$

Now lets find the expression of $\Delta g[b_t, m_t] = \frac{g(b_t) - g(m_t)}{g(m_t)}$ and $\Delta g[\bar{y}_t, m_t] = \frac{g(\bar{y}_t) - g(m_t)}{g(m_t)}$.

We use the relation $g(m_t) = \sum_{l=1}^4 q_l \epsilon_t^l + O(\epsilon_t^5)$ (i.e. a Taylor expansion of $g(m_t)$ around m^* till order 4). Using Taylor expansion of $g(b_t)$ and $g(\bar{y}_t)$ and after several algebraic manipulations we get

$$\begin{aligned} g(b_t) &= g(m_t + g(m_t)) \\ &= 0 + (\epsilon_t + g(m_t))q_1 + q_2(\epsilon_t + g(m_t))^2 \\ &\quad + q_3(\epsilon_t + g(m_t))^3 + c_4(\epsilon_t + g(m_t))^4 + O(\cdot) \end{aligned} \quad (\text{B.10})$$

$$\begin{aligned} g(\bar{y}_t) &= q_2(1 + q_1)\epsilon_t^2 + \frac{1}{q_1} [q_1(1 + q_1)(2 + q_1)q_3 \\ &\quad - q_1^2 q_2^2 - 2q_1 q_2^2 - 2q_2^2] \epsilon_t^3 \\ &\quad + \frac{\epsilon_t^4}{q_1^2} [(q_1^3 + 4q_1^2 + 7q_1 + 5)q_2^3 \\ &\quad - q_1 q_2 q_3 (2q_1^3 + 7q_1^2 + 10q_1 + 7) \\ &\quad + q_1^2 q_4 (1 + q_1)(q_1^2 + 3q_1 + 3)] + O(\epsilon_t^5) \end{aligned} \quad (\text{B.11})$$

□

Finally, plugging in the values of $g(\bar{y}_t)$, $g(m_t)$ and $g(b_t)$ back in equation (B.7) and after doing simplification we get

$$\epsilon_{t+1} = \frac{(1 + q_1)q_2[(5 + 3q_1)q_2^2 - q_1(1 + q_1)q_3]}{q_1^3} \epsilon_t^4 + O(\epsilon_t^5).$$

The above equation proves the claim and implies order four convergence. This completes the proof.

Proof. Proof of Proposition 9: Let $t \geq 2$. Using Proposition 8 the algorithm has a convergence order of 4. From Definition 12 we have $c_1 = \limsup_{t \rightarrow \infty} \frac{\eta_{t+1}}{\eta_t^4}$ which implies that $\eta_{t+1} \leq c_1 \eta_t^4$ which can be re written in terms of $\eta_{t+1} \leq c_1 \eta_t^4$

We reiterate the recursive equation

$$\eta_t \leq c_1 \eta_{t-1}^4 \quad (\text{B.12})$$

$$\leq c_1 (c_1 \eta_{t-2}^4)^4 = c_2^{1+4} \eta_{t-2}^{16} \quad (\text{B.13})$$

$$\leq c_1^{1+4} (c_1 \eta_{t-3}^4)^{16} \quad (\text{B.14})$$

$$= c_1^{1+4+4^2} \eta_{t-3}^{4^3} \quad (\text{B.15})$$

$$\leq c_1^{1+4+4^2+\dots+4^{t-1}} \eta_0^{4^t} \quad (\text{B.16})$$

We remember that for $q \neq 1$, $1 + q + \dots + q^{t-1} = \frac{q^t - 1}{q - 1}$. Thus,

$$\eta_t \leq c_1^{\frac{4^t - 1}{3}} \eta_0^{4^t}.$$

This means that the convergence time to be within an η -neighborhood of the mean-field equilibrium is at most for t satisfying $c_1^{\frac{4^t - 1}{3}} \eta_0^{4^t} \leq \eta$ where $c_1 = |c_2|$ whose value is given in the proof of Proposition 8. Thus,

$$(\eta_0 c_1^{\frac{1}{3}})^{4^t} \leq \eta c_1^{\frac{1}{3}}.$$

By taking the logarithm twice, one gets

$$T = \frac{1}{\ln(4)} \ln \left[\frac{\ln \left(\frac{1}{\eta c_1^{\frac{1}{3}}} \right)}{\ln \left(\frac{1}{\eta_0 c_1^{\frac{1}{3}}} \right)} \right],$$

which is the announced result. \square

Proof of Proposition 6: By assumption, the satisfaction levels s_i^* are such that the condition $\sum_{i=1}^n s_i^* < c_n$ is met. then we compute $s_i(a)$.

$$s_j(a) = \frac{c_n a_j^\alpha}{\sum_{j=1}^n a_j^\alpha} = \frac{c_n s_i^*}{\sum_{j=1}^n s_j^*} > s_i^*$$

Thus, every user is satisfied. \square

Bibliography

- [AAJW07] Vineet Abhishek, Sachin Adlakha, Ramesh Johari, and Gabriel Weintraub. Oblivious equilibrium for general stochastic games with many players. In *In Allerton Conference on Communications, Control and Computing*, 2007.
- [ABYST12] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafir. The resource-as-a-service (RaaS) cloud. In *USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '12)*, 2012.
- [AFA05] Carlos Als-Ferrer and Ana B. Ania. The evolutionary stability of perfectly competitive behavior. *Economic Theory*, 26(3):497–516, 2005.
- [AJ10] S. Adlakha and R. Johari. Mean field equilibrium in dynamic games with complementarities. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 6633–6638, Dec 2010.
- [AJG13] K.T. Atta, A. Johansson, and T. Gustafsson. On-line optimization of cone crushers using extremum-seeking control. In *Control Applications (CCA), 2013 IEEE International Conference on*, pages 1054–1060, Aug 2013.
- [APP11] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando. A game theoretic formulation of the service provisioning problem in cloud systems. In *Proceedings of the 20th international conference on World wide web, WWW '11*, pages 177–186, New York, NY, USA, 2011. ACM.
- [Ben99] Michel Benam. Dynamics of stochastic approximation algorithms. *Seminaire de probabilités de Strasbourg*, 33:1–68, 1999.

- [Ber10] D. P. Bertsekas. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. 2010.
- [BHN11] R. Bendlin, Yih-Fang Huang, and J.A. Nossek. Probabilistic power control for heterogeneous cellular networks with closed-access femtocells. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pages 1628–1632, 2011.
- [BJ12] P. Bianchi and J. Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non convex optimization. *Automatic Control, IEEE Transactions on*, PP(99):1, 2012.
- [Bor08] V.S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- [BTZ93] Michael R Baye, Guoqiang Tian, and Jianxin Zhou. Characterizations of the existence of equilibria in games with discontinuous and non-quasiconcave payoffs. *Review of Economic Studies*, 60(4):935–948, October 1993.
- [CA09] Vikram Chandrasekhar and Jeffrey G. Andrews. Uplink capacity and interference avoidance for two-tier femtocell networks. *Trans. Wireless. Comm.*, 8(7):3498–3509, July 2009.
- [DB74] G. Dahlquist and Å. Björck. *Numerical Methods*. Dover Books on Mathematics Series. Dover, 1974.
- [DGA14] E. Dincmen, B.A. Guvenc, and T. Acarman. Extremum-seeking control of abs braking in road vehicles with lateral force improvement. *Control Systems Technology, IEEE Transactions on*, 22(1):230–237, Jan 2014.
- [DMCS80] Pradeep Dubey, Andreau Mas-Colell, and Martin Shubik. Efficiency properties of strategies market games: An axiomatic approach. *Journal of Economic Theory*, 22(2):339–362, April 1980.
- [DSJE13] H.-B. Durr, M.S. Stankovic, K.H. Johansson, and C. Ebenbauer. Examples of distance-based synchronization: An extremum seeking approach. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 366–373, Oct 2013.

- [FB99] R.A. Fisher and H. Bennett. *The Genetical Theory of Natural Selection: A Complete Variorum Edition*. OUP Oxford, 1999.
- [FKB12a] P. Frihauf, M. Krstic, and T. Basar. Nash equilibrium seeking in noncooperative games. *Automatic Control, IEEE Transactions on*, 57(5):1192–1207, may 2012.
- [FKB12b] P. Frihauf, M. Krstic, and T. Basar. Nash equilibrium seeking in noncooperative games. *Automatic Control, IEEE Transactions on*, 57(5):1192–1207, may 2012.
- [FZRL08] I. Foster, Yong Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10, nov. 2008.
- [GDD13] M. Guay, S. Dhaliwal, and D. Dochain. A time-varying extremum-seeking control approach. In *American Control Conference (ACC), 2013*, pages 2643–2648, June 2013.
- [GKTH97] A. Ganz, C. M. Krishna, D. Tang, and Z. J. Haas. On optimal design of multitier wireless cellular systems. *Comm. Mag.*, 35(2):88–93, February 1997.
- [GLL11] Olivier Gueant, Jean-Michel Lasry, and Pierre-Louis Lions. In *Paris-Princeton Lectures on Mathematical Finance 2010*, volume 2003 of *Lecture Notes in Mathematics*. 2011.
- [GSF98] Michael A. Goodrich, Wynn C. Stirling, and Richard L. Frost. A theory of satisficing decisions and control. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 28(6):763–779, 1998.
- [GSS93] N.J. Gordon, D.J. Salmond, and A. F M Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.
- [Ham67] W. D. Hamilton. Extraordinary sex ratios. *Science*, 156(3774):477–488, April 1967.
- [HAT13] N.U. Hassan, M. Assaad, and H. Tembine. Distributed h^∞ -based power control in a dynamic wireless network environment. *Communications Letters, IEEE*, 17(6):1124–1127, 2013.

- [HTAZ12a] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Cloud networking mean field games. In *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pages 46–50, 2012.
- [HTAZ12b] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Cloud networking mean field games. In *Cloud Networking (CloudNet), 2012 IEEE 1st International Conference on*, nov 2012.
- [HTAZ12c] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed stochastic learning for continuous power control in wireless networks. In *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*, pages 199–203, 2012.
- [HTAZ13a] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed power control in femto cells using bayesian density tracking. In *51st Annual Allerton Conference on Communication, Control and Computing*, 2013.
- [HTAZ13b] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Distributed transmit beamforming with 1-bit feedback for los-miso channels. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pages 455–459, 2013.
- [HTAZ13c] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. Mean-field games for resource sharing in cloud based networks. *Submitted to IEEE Transactions on Networking*, 2013.
- [HTAZ13d] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. On the convergence of a nash seeking algorithm with stochastic state dependent payoffs. *Submitted to IEEE Transactions on Automatic Control*, 2013.
- [HTAZ13e] A.F. Hanif, H. Tembine, M. Assaad, and D. Zeghlache. On the convergence of a nash seeking algorithm with stochastic state dependent payoffs. *Submitted to IEEE Transactions on Automatic Control*, 2013.

- [Hua12] Mean field stochastic games with discrete states and mixed players. In Vikram Krishnamurthy, Qing Zhao, Minyi Huang, and Yonggang Wen, editors, *Game Theory for Networks*, volume 105 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. 2012.
- [HYC09] Eun Jin Hong, Se Young Yun, and Dong-Ho Cho. Decentralized power control scheme in femtocell networks: A game theoretic approach. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 415–419, 2009.
- [IDME12] A.S. Ibrahim, A.M. Darwish, E.A. Makled, and H. Elgebaly. Adaptive interference mitigation techniques for femtocells. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 1218–1223, 2012.
- [Inc08] Amazon Inc. *Amazon Elastic Compute Cloud (Amazon EC2)*. Amazon Inc., <http://aws.amazon.com/ec2/#pricing>, 2008.
- [Ish74] Shiro Ishikawa. Fixed points by a new iteration method. *Proc. Am. Math. Soc.*, 44:147–150, 1974.
- [Jr.50] John F. Nash Jr. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, January 1950.
- [KA06] S.U. Khan and I. Ahmad. Non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 10 pp., april 2006.
- [KDF⁺11] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki. Optimal service pricing for a cloud cache. *Knowledge and Data Engineering, IEEE Transactions on*, 23(9):1345–1358, sept. 2011.
- [KK12] Jörn Kunsemöller and Holger Karl. A game-theoretical approach to the benefits of cloud computing. In *Proceedings of the 8th international conference on Economics of Grids, Clouds, Systems*,

- and Services*, GECON'11, pages 148–160, Berlin, Heidelberg, 2012. Springer-Verlag.
- [KMNT12] Sei Zhen Khong, C. Manzie, D. Nesic, and Ying Tan. Multidimensional global extremum seeking via the direct method. In *Control Conference (AUCC), 2012 2nd Australian*, pages 277–282, Nov 2012.
- [KSL07] E. Karipidis, N.D. Sidiropoulos, and Zhi-Quan Luo. Far-field multicast beamforming for uniform linear antenna arrays. *Signal Processing, IEEE Transactions on*, 55(10):4916–4927, oct. 2007.
- [Kus03] Yin G.G Kushner, H.J. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.
- [KW52] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23:462–466, 1952.
- [LK10] Shu-Jun Liu and M. Krstic. Stochastic averaging in continuous time and its applications to extremum seeking. *Automatic Control, IEEE Transactions on*, 55(10):2235–2250, oct. 2010.
- [LK11] Shu-Jun Liu and Miroslav Krstic. Stochastic nash equilibrium seeking for games with general nonlinear payoffs. *SIAM J. Control and Optimization*, 49(4):1659–1679, 2011.
- [LK12] Shu-Jun Liu and M. Krstic. Newton-based stochastic extremum seeking. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 4449–4454, Dec 2012.
- [MCL⁺10] Richard T.B. Ma, Dah Ming Chiu, John C.S. Lui, Vishal Misra, and Dan Rubenstein. On resource management for cloud users: A generalized kelly mechanism approach. Technical report, Electrical Engineering, May 2010.
- [MP73] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

- [NB01] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. on Optimization*, 12(1):109–138, January 2001.
- [NMM13] D. Nedic, A. Mohammadi, and C. Manzie. A framework for extremum seeking control of systems with parameter uncertainties. *Automatic Control, IEEE Transactions on*, 58(2):435–448, Feb 2013.
- [NNTM12] D. Nedic, T. Nguyen, Y. Tan, and C. Manzie. On a shubert algorithm-based global extremum seeking scheme. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1615–1620, Dec 2012.
- [NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press New York, NY, USA ISBN:0521872820, 2007.
- [NTM⁺12] D. Nedic, Ying Tan, C. Manzie, A. Mohammadi, and W. Moase. A unifying framework for analysis and design of extremum seeking controllers. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 4274–4285, May 2012.
- [PB11a] N. Parikh and S. Boyd. Block splitting for large-scale distributed learning. In *Neural Information Processing Systems (NIPS) Workshop on Big Learning*, 2011.
- [PB11b] N. Parikh and S. Boyd. Cloud networking: A network approach that meets the requirements of cloud computing. In *CQ2*, 2011.
- [Pos03] Alex Possajennikov. Evolutionary foundations of aggregate-taking behavior. *Economic Theory*, 21(4):921–928, 2003.
- [PQ12a] J. Poveda and N. Quijano. Dynamic bandwidth allocation in wireless networks using a shahshahani gradient based extremum seeking control. In *Network Games, Control and Optimization (Net-GCooP), 2012 6th International Conference on*, pages 44–50, Nov 2012.

- [PQ12b] J. Poveda and N. Quijano. A shahshahani gradient based extremum seeking scheme. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5104–5109, Dec 2012.
- [PQ13] J. Poveda and N. Quijano. Distributed extremum seeking for real-time resource allocation. In *American Control Conference (ACC), 2013*, pages 2772–2777, June 2013.
- [Ran10] S. Rangan. Femto-macro cellular interference control with subband scheduling and interference cancelation. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 695–700, 2010.
- [RM51] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.
- [RN04] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks, IPSN '04*, pages 20–27, New York, NY, USA, 2004. ACM.
- [Rou12] Margaret Rouse. Spi model (saas, paas, iaas), February 2012.
- [RY12] S. Poole C. Ma F. He J. Zhuang Rao, N. and D. Yau. Cloud computing infrastructure robustness: A game theory approach. In *Proceedings of the International Conference on Computing, Networking and Communications*, 2012.
- [Sch88] M. E. Schaffer. Evolutionarily stable strategies for a finite population and a variable contest size. *Journal of theoretical biology*, 132(4):469–478, June 1988.
- [SG99] Wynn C. Stirling and Michael A. Goodrich. Satisficing games. *Inf. Sci.*, 114(1-4):255–280, 1999.
- [Sim56] H. A. Simon. Rational choice and the structure of the environment. *Psychological Review; Psychological Review*, 63(2):129–138, 1956.
- [SJS12] M.S. Stankovic, K.H. Johansson, and D.M. Stipanovic. Distributed seeking of nash equilibria with applications to mobile sensor networks. *Automatic Control, IEEE Transactions on*, 57(4):904–919, april 2012.

- [Sny05] J. A. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Applied Optimization, Vol. 97. Springer-Verlag New York, Inc., second edition, 2005.
- [SR64] L. Schwartz and P.H. Rabinowitz. *Functional analysis*. Courant Institute Lecture Notes. Courant Institute of Mathematical Sciences, New York University, 1964.
- [SS09] M.S. Stankovic and D.M. Stipanovic. Stochastic extremum seeking with applications to mobile sensor networks. In *American Control Conference, 2009. ACC '09.*, pages 5622–5627, june 2009.
- [SS10] Miloš S. Stanković and Dušan M. Stipanović. Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica*, 46(8):1243–1251, August 2010.
- [Sti03] W.C. Stirling. *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge University Press, 2003.
- [TBA86] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *Automatic Control, IEEE Transactions on*, 31(9):803–812, sep 1986.
- [Tem12] H Tembine. Distributed strategic learning for wireless engineers. *CRC Press, Taylor & Francis Inc., ISBN 9781439876374, 496 pages*, 2012.
- [TM10] Fei Teng and Frédéric Magoulès. A new game theoretical resource allocation algorithm for cloud computing. In *Proceedings of the 5th international conference on Advances in Grid and Pervasive Computing, GPC'10*, pages 321–330, Berlin, Heidelberg, 2010. Springer-Verlag.
- [Tul80] Gordon Tullock. *Efficient Rent Seeking*, pages 97–112. Texas A&M University Press, College Station, TX, 1980.
- [vdWHN12] N. van de Wouw, M. Haring, and D. Nesic. Extremum-seeking control for periodic steady-state response optimization. In *Decision*

- and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1603–1608, Dec 2012.
- [VJ10] Indranil Gupta Matthew Caesar Virajith Jalaparti, Giang Nguyen. Cloud resource allocation games. Technical report, Dept of Computer Science, Dec 2010.
- [Vol] V. Volterra. Variations and fluctuations of the number of individuals in animal species living together. *Mem. Acad. Lincei Roma*, 2:31–113.
- [War52] J.G. Wardrop. *Wardrop of Some Theoretical Aspects of Road Traffic Research-Road Paper*. Number no. 36. Road Engineering Division, 1952.
- [WBJ12] Yu Wu, Loc Bui, and R. Johari. Heavy traffic approximation of equilibria in resource sharing games. *Selected Areas in Communications, IEEE Journal on*, 30(11):2200–2209, December 2012.
- [WCLP12] J.Y. Wu, Xiaoli Chu, and D. Lopez-Perez. Downlink outage probability of co-channel femtocells in hierarchical 3-sector macrocells. *Communications Letters, IEEE*, 16(5):698–701, 2012.
- [Wil70] Stephen Willard. *General topology*. Addison-Wesley, Reading, MA, 1970.
- [WVZX10] Guiyi Wei, Athanasios V. Vasilakos, Yao Zheng, and Naixue Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *J. Supercomput.*, 54(2):252–269, November 2010.
- [XCA10] Ping Xia, V. Chandrasekhar, and J.G. Andrews. Open vs. closed access femtocells in the uplink. *Wireless Communications, IEEE Transactions on*, 9(12):3798–3809, 2010.
- [Yat95] R.D. Yates. A framework for uplink power control in cellular radio systems. *Selected Areas in Communications, IEEE Journal on*, 13(7):1341–1347, 1995.
- [YH13] Maojiao Ye and Guoqiang Hu. Extremum seeking under input constraint for systems with a time-varying extremum. In *Decision*

- and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1708–1713, Dec 2013.
- [YW11] Lin Yang and Pingping Wen. Location based autonomous power control for icic in lte-a heterogeneous networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6, 2011.
- [YZ11] Tao Yang and Lu Zhang. Approaches to enhancing autonomous power control at femto under co-channel deployment of macrocell and femtocell. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, pages 71–75, 2011.
- [ZG12] Yinghua Zhang and N. Gans. Simplex guided extremum seeking control for real-time optimization. In *American Control Conference (ACC), 2012*, pages 3377–3382, June 2012.