Master's Thesis

Thesis Template That You Want

Sunho Kim

Department of Electrical and Computer Engineering

Graduate School
Korea University

February 2023

Thesis Template
That You Want

by

Sunho Kim

under the supervision of Professor Sang Hyun Lee

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Graduate School
Korea University

November 2022

The thesis of Sunho Kim has been approved
by the thesis committee in partial fulfillment of
the requirements for the degree of
Master of Science

November 2022

Committee Chair: Referee 1

Committee Member: Referee 2

Committee Member: Referee 3

Thesis Template
That You Want

by Sunho Kim

Department of Electrical and Computer Engineering

under the supervision of Professor Sang Hyun Lee

Abstract

Please write an English abstract here.

Keywords: Korea University, thesis, dissertation, organizing, formatting

당신이 원하던 템플릿

김 선 호

전 기 전 자 공 학 과

지도교수: 이 상 현

초록

이곳에 국문 초록을 작성하여 주십시오.

주제어: 고려대학교, 학위논문, 작성방법

감사의 글 예시 페이지입니다.

감사합니다.

Preface

This thesis is submitted for the degree of Master of Science at the Korea University. All of the work presented henceforth has been conducted in the *Lab for Informatics, Communications, and Systems* at the Korea University, under the supervision of professor Sang Hyun Lee. The research described herein was conducted under the supervision of professor Sang Hyun Lee, between October 1998 and October 2001. It is to the best of my knowledge original, except where acknowledgments and references are made to previous work.

Part of this thesis has been submitted as a conference paper and accepted for presentation in IEEE GLOBECOM 2022 Workshops. The corresponding information is as follows :

S. Kim, H. K. Kim, and S. H. Lee, “Survey Propagation for Cell-Free Massive MIMO Pilot Assignment,” in *Proc. IEEE Globecom Workshops (GC Wkshps), Rio de Janeiro, Brazil, Dec. 2022*, in press.

While the project has been performed by my lead, there have been thankful contributions from several intellectuals. Professor Sang Hyun Lee is the supervisory author of the paper and has been involved throughout the project. His insights and knowledge have motivated to employ a physical approach in this work. Hong Ki Kim has organized a simulation environment for the demonstration of our algorithm, and co-worked on manuscript edits and proofreading. Hong Ki Kim established a simulation environment for the verification of our proposed algorithm, and then worked together on editing and reviewing the manuscript. I sincerely appreciate their true support.

Acknowledgement

이 패키지는 기존 https://github.com/KUNPL/KUThesis의 양식을 사용하여 고려대학교 학위논문 작성법이 요구하는 요소를 만족하도록 작성하였습니다. 석사/박사 영문논문 작성을 전제로 되도록 필요한 요소만을 남겨놓았으며, 대부분 주석을 통해 활용법을 안내하였으나, 기본적인 사용법을 아래에 설명드립니다.

*********** 사용법 ************* .

[박사과정의 경우]

1. 박사과정의 경우 "myThesisTemp.cls" 파일에서 line 20 23를 주석의 안내에 맞춰 수정한다.

[공통 사용법]

0. 컴파일러는 pdfLaTex 2014(legacy)로 설정하는 것을 권장합니다.

1. main.tex에서 논문제목, 이름, 지도교수명, 심사위원명단 등의 사항을 입력한다.

2. 초록, 서문, 감사의 글 등은 "beginnings" 폴더에 있는 각 파일에서 수정한다.

3. 본문의 경우 "chapters" 폴더에 있는 파일들을 참고해서 작성한다.

4. 본문 관련 tex파일을 추가하는 경우, "main.tex"파일에

꼭 \addContents{chapters/~~~}을 추가한다.

5. 참고문헌은 "library.bib" 파일에 저장한다.

6. 감사의 글, 서문, 사사 포함여부에 따라 "myThesisTemp.cls" 파일에서 line 277,279,281의 주석처리 여부를 결정한다.

7. 논문에 도식, 표 등의 포함여부에 따라 "myThesisTemp.cls" 파일에서 line 572,573의 주석처리 여부를 결정한다.

[기타]

1) \bibitem을 사용하여 Bibliography을 채우시는 분은 bibliography.tex를 따로 생성하여(이름은 바꿔도 상관없음) 그 파일에 인용 문헌을 적으시고 myThesisTemp.cls 파일 안의 \bibliography{library.bib} 대신

```
\begin{thebibliography}{}
\input{bibliography.tex}
\end{thebibliography}
```

를 입력하시면 됩니다.

2) Theorem, Lemma 등을 이용하고 싶으면

```
\newtheorem{theorem}{Theorem}
\newtheorem{definition}{Definition}
\newtheorem{lemma}{Lemma}
\newtheorem{property}{Property}
```

myThesisTemp.cls 파일 안의 Preamble 위에(아무데나 써도 상관은 없을테지만) 입력하면 됩니다.

3) proof을 이용하고 싶으면 `\RequirePackage`에 amsthm를 추가하면 됩니다.

4) appendix를 맨 마지막에 추가하려면 myThesisTemp.cls 파일 안의 bibliography 다음에

```
\appendix
\input{appendices.tex}
\addevenpage
```

를 입력하시면 됩니다.

그 후 appendices.tex를 따로 생성하여 각 appendix를 `\chapter`로 지정해주면 알파벳 형식 appendix가 나옵니다.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cell-free (CF) massive multiple-input multiple-output (MIMO) technology has been spotlighted as one of the most promising architectures for future wireless networks [1, 2, 3, 4]. In the CF MIMO transmission scenario, massive-scale of access points (APs) are spatially distributed across the coverage to jointly serve users with shared radio resources as illustrated in Fig. 1.1.

1.1 Motivation

The quality-of-service with improved user fairness can be enhanced by the macro-diversity of CF massive MIMO techniques [5]. For the full potential of this concept, the qualified channel state information (CSI) from channel estimation based on pilot training is of utmost importance [6]. However, the elimination of cell boundaries incurs a main trade-off challenge in the pilot usage (see Fig. 1.2); The pilot reuse is unavoidable due to insufficient sequence diversity for orthogonal pilot assignment (PA) over the coverage [7]. Non-orthogonal PA among pilot-sharing users is subject to the pilot contamination, degrading the channel estimation accuracy and the subsequent system performance [6]. A proper strategy is essential for alleviation of the contamination and improvement in network

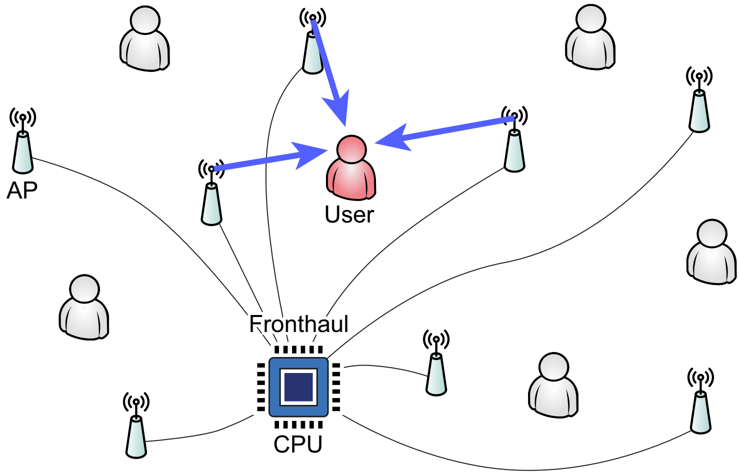


Figure 1.1: Cell-free massive MIMO system.

throughput.

Despite the urgent need for the optimal assignment of the pilot sequence, a well-structured PA framework remains yet partially addressed, and the associated combinatorial challenges are found computationally demanding [8]. Furthermore, the resulting throughput gain to a group of pilot-sharing users is hard to address. A variety of PA frameworks have been investigated for this challenge. A simple approach sticks to a greedy policy that begins with a random assignment and repeated modification of the worst case user for gradual throughput performance improvement [1]. An iterative user selection is developed to assign orthogonal pilots via bipartite matching implemented using Hungarian algorithms [9]. In [10], a weighted graphic framework have been proposed that the contamination factors defined between two UEs are used to assignment. The usage of a variant of the hierarchical agglomerative clustering algorithm has been suggested in [11]. There have also been attempts to apply methods such as tabu-search [12] or graph coloring [13] to PA in CF massive MIMO. Furthermore, a topological interference management is considered in sparse connectivity cases [8]. Although a structured PA framework is also proposed using K-means clustering, geographic information

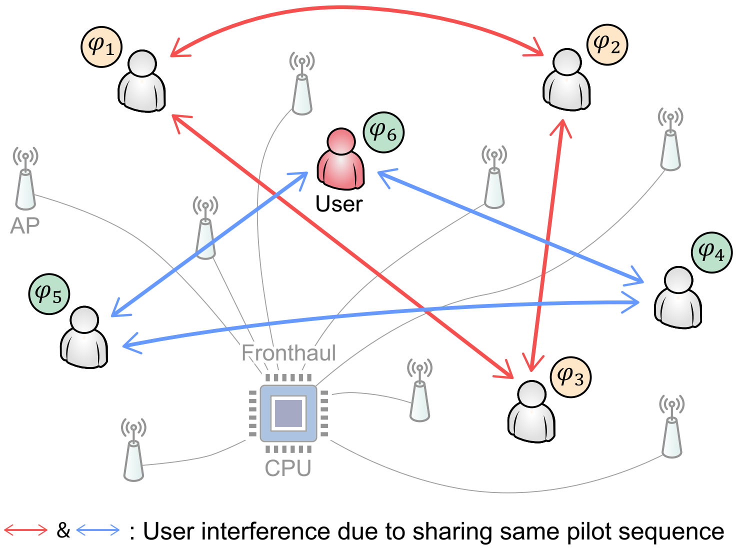


Figure 1.2: Pilot reuse problem in CF massive MIMO.

is used instead of throughput measurements [4].

1.2 Contributions

This work develops a new optimization framework for the CF massive MIMO PA task. While existing works require a completely new PA on every coherence interval, we consider some succession of previous assignments for sustainable PA. In this regard, the maximization of the network utility is conducted under the pre-assigned states for some users. The resulting optimization task is formulated into a special class of quadratic assignment problems [14], which fall in a class of *NP*-hard problems [15]. The PA challenge entails group-wise matching where fixing an assignment of a group enables or disables assignments of many other groups that share the same users. The PA solution space is thus separated into interacting *clusters* of feasible configurations, thereby rendering the PA search quickly demanding as the system scales up.

To tackle this, a novel approach is developed by borrowing the principles of survey propagation (SP) that has originally been derived to address the interaction relationship among multiple particles from statistical physics [16, 17]. Based on this formalism, multiple states of pilot to user-group association can be efficiently described and solved using a distributed algorithm that obtains an efficient solution. Thus, the proposed distributed algorithm inspired by survey propagation accomplishes a comprehensive search over the clusters of PA solution space without brute-force enumeration.

1.3 Thesis Outline and Notations

The remainder of this thesis is organized as follows. The system model of CF massive MIMO and the main metric are first introduced. In chapter 3, the corresponding PA problem is formulated in combinatorial way. Our proposed framework is provided in chapter 4. The design principles and derivations of SP algorithm are also described in this chapter. The numerical results in chapter 5 demonstrate that the network throughput is consistently achieved higher than existing techniques. Finally, the major conclusion is suggested in chapter 6.

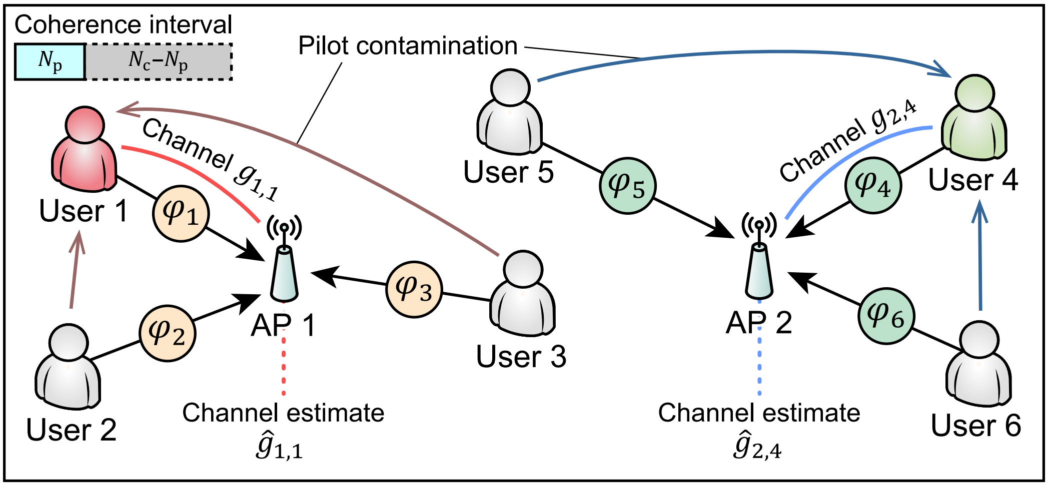
Throughout this thesis, a boldface lowercase letter, x, denote column vectors. The superscripts $*$ and H denote conjugate and conjugate transpose, respectively. The expected value of $\mathbf{x}$ is denoted as $\mathbb{E}\{\mathbf{x}\}$. $\mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R})$ denotes the multivariate circularly symmetric complex Gaussian distribution with correlation matrix $\mathbf{R}$. Other mathmetical notations in some expressions are described in the first part where they first appear.

Chapter 2

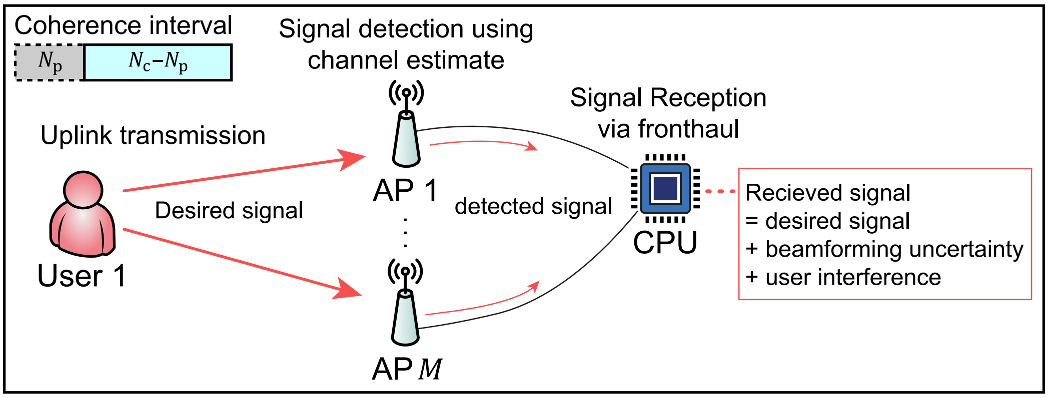
Cell-Free Massive MIMO System

A CF massive MIMO system consisting of M APs and K users ($K \leq M$), all equipped with a single antenna, is considered. Each user is served simultaneously by multiple APs operating with the shared radio resource. The coherence intervals of all channels are thus defined as the product of the coherence time and bandwidth. Individual APs are connected to a central processing unit (CPU) via fronthaul that provides error-free and infinite capacity links for straightforward focus on the PA operation.

This work mainly considers the uplink transmission since there is no need for downlink pilot training [1] and the channel reciprocity allows a straightforward extension to the downlink transmission [4]. In the uplink transmission operating in a time division duplex manner, a coherence interval, with its resource length N_{c}, is divided into two phases: the uplink pilot training and the uplink payload transmission. During the first training phase of length N_{p}, users transmit dedicated pilot sequences to APs, and APs attempt to estimate the user channel states. In the second phase of length $N_{\mathrm{c}} - N_{\mathrm{p}}$, the uplink signals are detected with those estimates.



(a) Uplink training phase with pilot reuse



(b) Uplink payload transmission phase

Figure 2.1: Uplink transmission in cell-free massive MIMO system.

2.1 Uplink Training

Let $\boldsymbol{\varphi}_k \in \mathbb{C}^{N_{\mathrm{p}} \times 1}$ be the unit norm pilot sequence used by k-th user, where N_{p} of orthogonal pilot sequences are available. The channel coefficient between the k-th user and the m-th AP during a coherence interval, denoted as g_{mk}, is defined by the small-scale fading h_{mk} and large-scale fading β_{mk} of the corresponding user-AP pair as $g_{mk} = \sqrt{\beta_{mk}} h_{mk}$. We assume that the entries of h_{mk} are independent and identically distributed (i.i.d) $\mathcal{N}_{\mathbb{C}}(0,1)$ random variables (RVs), and the channels are constant during the coherence interval. The received pilot signal at m-th AP is given by

$$\mathbf{y}_m = \sqrt{\omega N_{\mathrm{p}}} \sum_{k=1}^{K} g_{mk} \boldsymbol{\varphi}_k + \mathbf{n}_m \,, \tag{2.1}$$

where ω is the normalized transmit signal-to-noise ratio (SNR) of a pilot signal, and $\mathbf{n}_m$ is an additive noise vector of the m-th AP such that the elements of $\mathbf{n}_m$ are i.i.d. $\mathcal{N}_{\mathbb{C}}(0,1)$ RVs. The estimate of g_{mk} obtained based on a minimum-mean square error is given by

$$\hat{g}_{mk} = \frac{\mathbb{E}\left\{ \left(\boldsymbol{\varphi}_k^H \mathbf{y}_m\right)^* g_{mk} \right\}}{\mathbb{E}\left\{ \left|\boldsymbol{\varphi}_k^H \mathbf{y}_m\right|^2 \right\}} \left(\boldsymbol{\varphi}_k^H \mathbf{y}_m\right) = c_{mk} \left(\boldsymbol{\varphi}_k^H \mathbf{y}_m\right), \tag{2.2}$$

where

$$c_{mk} \triangleq \frac{\sqrt{N_{\mathrm{p}} \omega} \beta_{mk}}{N_{\mathrm{p}} \omega \sum_{k'=1}^{K} \beta_{mk'} \left|\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'}\right|^2 + 1} \,. \tag{2.3}$$

From the randomness assumptions for corresponding RVs, the mean-square channel estimate, denoted as γ_{mk}, is given by

$$\gamma_{mk} \triangleq \mathbb{E}\left\{ |\hat{g}_{mk}|^2 \right\} = \frac{N_{\mathrm{p}} \beta_{mk}^2}{N_{\mathrm{p}} \sum_{k'=1}^{K} \beta_{mk'} \left|\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'}\right|^2 + 1/\omega} \,. \tag{2.4}$$

The orthogonal PA leads to $\left|\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'}\right|^2 = 0$ if $k' \neq k$. The number of users K generally exceeds the number of orthogonal pilot resources N_{p}, and the same pilot sequences are necessarily reused in the assignment[1, 18]. The quality of the channel estimate $\hat{g}_{mk}$ is degraded by the signal transmitted from a pilot-sharing user k' with $\boldsymbol{\varphi}_{k'} = \boldsymbol{\varphi}_k$. This induces the pilot contamination. Fig. 2.1(a) illustrates a simple example. Two pilot resources are assigned to three users. While AP 1 estimates the channel for user 1, pilot signals from users 2 and 3 interfere due to the pilot reuse.

2.2 Uplink Payload Data Transmission

In uplink payload transmission from users in Fig. 2.1(b), all K users simultaneously send their data to the APs, and each AP detects the signals from channel estimates and sends them to the CPU. According to an analysis in [1], the received signal collected at the CPU can be decomposed into the desired signal (DS) component, the beamforming uncertainty (BU) component, the user interference (UI) component, and noise. Their average powers of k-th user are given, respectively, by

$$\mathrm{DS}_k = \omega\eta_k \left(\sum_{m=1}^{M} \gamma_{mk}\right)^2, \tag{2.5a}$$

$$\mathrm{BU}_k = \omega \sum_{k'=1}^{K} \eta_{k'} \sum_{m=1}^{M} \gamma_{mk}\beta_{mk'} , \tag{2.5b}$$

$$\mathrm{UI}_{kk'} = \omega\eta_{k'} \left(\sum_{m=1}^{M} \gamma_{mk} \frac{\beta_{mk'}}{\beta_{mk}}\right)^2 \left|\boldsymbol{\varphi}_k^H \boldsymbol{\varphi}_{k'}\right|^2, \tag{2.5c}$$

where $\sqrt{\eta_k}$ is a power control coefficient of the k-th user, i.e., $\eta_k \in (0, 1]$. We assume that the normalized uplink transmit SNR is equal to that of pilot signals, ω. Note that (2.5c) corresponds to the degradation effect of the pilot reuse on an achievable rate of k-th user in case of $k \neq k'$. The signal to interference and noise ratio (SINR) of k-th user and the corresponding achievable rate are given, respectively, by

$$\mathrm{SINR}_k = \frac{\mathrm{DS}_k}{\mathrm{BU}_k + \sum\limits_{k' \neq k}^{K} \mathrm{UI}_{kk'} + \sum\limits_{m=1}^{M} \gamma_{mk}} \ , \tag{2.6a}$$

$$R_k = \left(1 - \frac{N_{\mathrm{p}}}{N_{\mathrm{c}}}\right) \log_2 \left(1 + \mathrm{SINR}_k\right) \ . \tag{2.6b}$$

The network throughput maximization is the main purpose of this work. In the next chapter, the PA optimization task is formulated with the achievable rate for each user.

Chapter 3

Pilot Assignment Problem

The achievable rate for each user is damaged by the interference among pilot-sharing users. In this regard, the PA management is required constantly to ensure the network performance. While the pilot resources are repeatedly managed for each coherence interval, the large-scale channel parameters for some users are assumed to be preserved even if the interval is updated. In order to alleviate the computational loads, the latest assignments for those cases can leverage the non-orthogonal PA task. The initial setup policy is conducted so that the results of previous assignments for static users are maintained in the new PA. To this end, such users are preassigned with the same pilot sequence as the previous assignment, and it is assumed that N_p of users are orthogonally preassigned to all of the pilot resources.

For ease of description, a simple example in Fig. 3.1 is considered where two pilots are assigned to six users and both are preassigned to user 1 and user 4, respectively. Let a set $\mathcal{P}_a$ be indices of preassigned users to which the a-th pilot sequence is dedicated, i.e., $\mathcal{P}_1 = \{1\}$ and $\mathcal{P}_2 = \{4\}$. The PA problem is characterized by a combinatorial optimization to select pilot-sharing groups that share the same pilot sequences as the preassigned users. The PA solution is obtained so that the number of pilot-sharing users is as uniform as possible over pilot sequences.

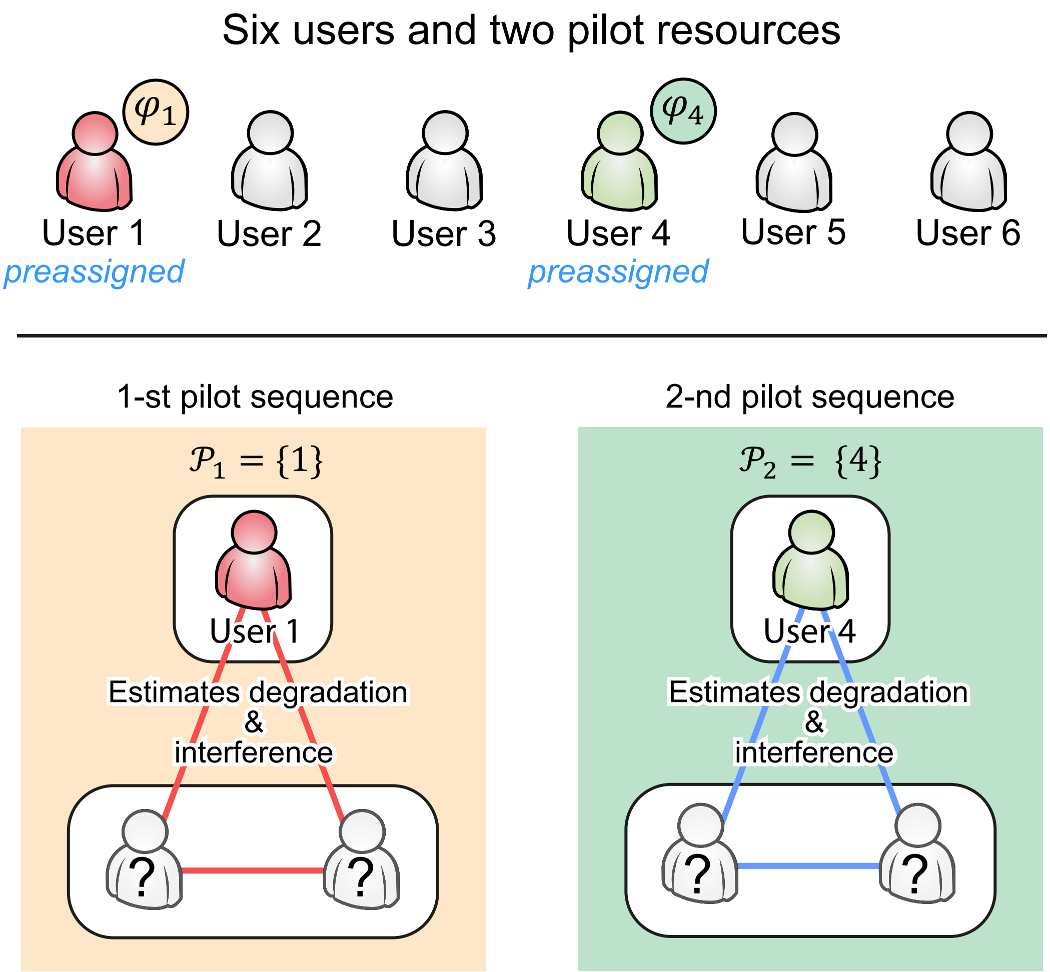


Figure 3.1: An example of 6-user PA with preassignment.

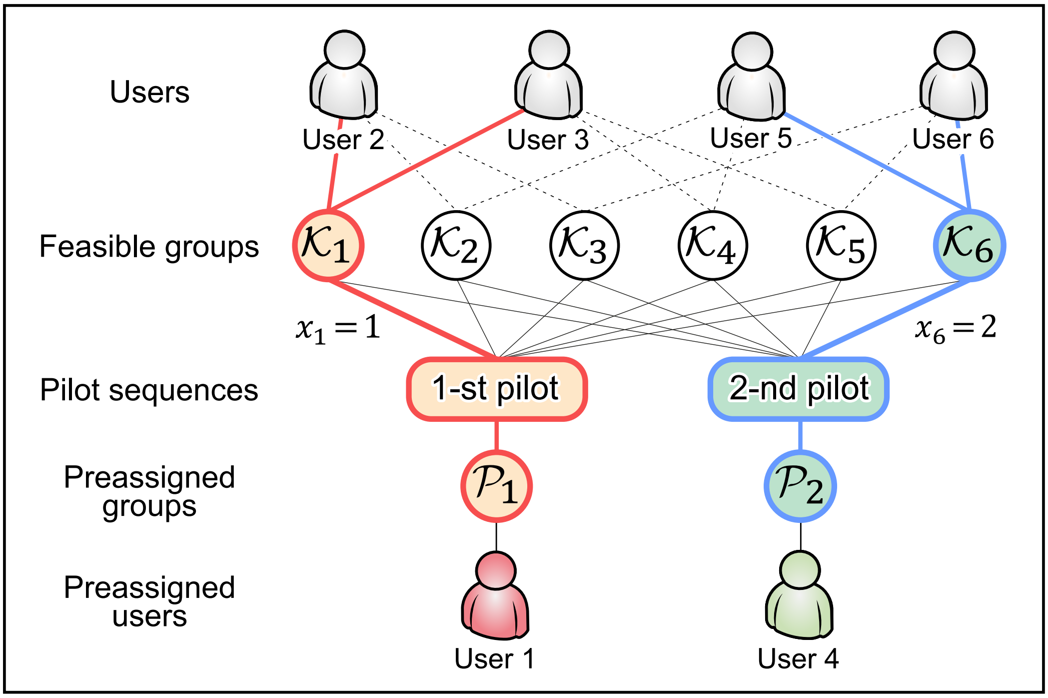


Figure 3.2: The relationship among variables in a 6-user PA example.

Since $K - N_p$ remaining users need the assignment, it is grouped into $(K - N_p)/N_p$ members, and there are $\binom{K-N_p}{K/N_p-1}$ different feasible groups. Let $\mathcal{K}_i$ denote the i-th feasible group ($i = 1, \ldots, \binom{K-N_p}{K/N_p-1}$). In the example, we have $\binom{4}{2} = 6$ feasible groups, i.e., $\mathcal{K}_1 = \{2, 3\}, \ldots, \mathcal{K}_6 = \{5, 6\}$. Since each user is assigned a single pilot, the feasible groups are selected so that the elements do not overlap in one solution. For the concrete indication of the PA, an integer variable x_i is associated with the pilot sequence assigned to users in group $\mathcal{K}_i$. Thus, It holds that $x_i = a$ if the group $\mathcal{K}_i$ is matched with a-th pilot and $x_i = 0$ otherwise. In addition, the corresponding state vector $\mathbf{x} = [x_1, \ldots, x_6]$ is defined. Fig. 3.2 shows the example for the case where $\mathcal{K}_1$ and $\mathcal{K}_6$ are used for PA among the feasible groups. The corresponding solution is $\mathbf{x} = [1, 0, 0, 0, 0, 2]$.

Among the valid states, PA tasks are purposed to find the optimal solution that maximizes the network-wide utility such as sum-rate. A partial sum-rate denoted by R_{ia} is defined corresponding to the pilot-sharing members if users in $\mathcal{K}_i$ are assigned to the a-th pilot and given as

$$R_{ia} \triangleq \sum_{k \in \mathcal{K}_i} R_k + \sum_{l \in \mathcal{P}_a} R_l \,. \tag{3.1}$$

3.1 Generic Formulation

Let $\mathcal{I}$ denote the set of indices for feasible groups and $\mathcal{A} = \{1, \ldots, N_{\mathrm{p}}\}$ be the set for pilot resources. the corresponding PA problem is formulated as

$$\textbf{(P1)} \;\; \max_{\mathbf{x}} \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}} R_{ia} \mathbb{I}\left(x_i = a\right) \tag{3.2a}$$

$$\text{subject to } x_i \in \mathcal{A} \cup \{0\}\,,\; \forall i \in \mathcal{I}\,, \tag{3.2b}$$

$$\sum_{i \in \mathcal{I}} \mathbb{I}\left(x_i = a\right) \leq 1\,,\; \forall a \in \mathcal{A}\,, \tag{3.2c}$$

$$\sum_{i' \in \mathcal{N}(i)} \mathbb{I}\left(\left(x_i \neq 0\right) \wedge \left(x_{i'} \neq 0\right)\right) = 0\,,\; \forall i \in \mathcal{I}\,, \tag{3.2d}$$

where $\mathbb{I}(\cdot)$ is the indicator function that yields one if the input statement holds true and zero otherwise, $\wedge$ is the logical intersection, and $\mathcal{N}(i)$ is the set of neighborhoods of the i-th group, i.e., $\mathcal{N}(i) = \{i' : \mathcal{K}_i \cap \mathcal{K}_{i'} \neq \emptyset, i' \neq i\}$. The constraint in (3.2c) indicates that at most one group is allocated to each pilot. Furthermore, the constraint in (3.2d) means that feasible groups containing the same users are not used simultaneously for a PA solution.

3.2 Unconstrained Optimization Formulation

The constraints in **(P1)** are associated with the groups assigned to individual pilots. This point of view motivates the introduction of a new function that considers the entire constraints. To represent them in an analytical way, a penalizing function for a-th pilot, $\mathcal{F}_a(\mathbf{x})$, is defined to return a negative infinite value when the input vector $\mathbf{x}$ violates constraints related to (3.2c) and (3.2d) as

$$\mathcal{F}_a(\mathbf{x}) \triangleq \begin{cases} -\infty & \text{if } \sum_{i\in\mathcal{I}} \mathbb{I}(x_i = a) > 1\,, & \text{(3.3a)} \\ & \text{or } \exists i \text{ s.t. } \sum_{i'\in\mathcal{N}(i)} \mathbb{I}((x_i \neq 0) \wedge (x_{i'} \neq 0)) \neq 0\,, & \text{(3.3b)} \\ 0 & \text{otherwise}\,. & \text{(3.3c)} \end{cases}$$

Note that the case (3.3a) is associated with the constraint in (3.2c), but specifically focuses on the a-th pilot. The case (3.3b) corresponds to that some neighboring groups are used in a PA solution so violating the constraint in (3.2d). Likewise, a utility function $\mathcal{R}_i$ is defined to represent the weight for the objective function by the value of variable x_i as

$$\mathcal{R}_i(x_i) \triangleq \sum_{a\in\mathcal{A}} R_{ia}\mathbb{I}(x_i = a)\,. \tag{3.4}$$

An unconstrained optimization formulation from **(P1)** is obtained from (3.4) as

$$\textbf{(P2)}\ \max_{\mathbf{x}} \sum_{i\in\mathcal{I}} \mathcal{R}_i(x_i) + \sum_{a\in\mathcal{A}} \mathcal{F}_a(\mathbf{x})\,. \tag{3.5}$$

The optimal solution of (**P2**) is identical to that of (**P1**). Since this is factorized as a simple sum of utility functions and penalizing functions, this allows us to establish a distributed approach to solve the problem. The optimal states are found locally by individual functions, and their information is exchanged for comparison with each other. The state variables are repeatedly corrected to deactivate all the penalizing functions and are expected to converge to a global solution that optimizes the total sum of the utility functions. In the next chapter, we presents an efficient solution for **(P2)** based on a cooperative consideration over a graphical representation of the overall optimization task.

Chapter 4

Distributed Algorithm via Survey Propagation

It is noted that the PA problem inherits a unique nature that states interactively constrain each other, thus incurring many local solutions that significantly differ from the optimum. This corresponds to so-called *replica symmetry breaking* phenomenon in the field of statistical physics [19]. The SP [16] approach is investigated to tackle this challenge induced by such property. From the analogy of the PA solution space with the formation of stable interaction relationships among variables, this approach can be exploited to characterize an efficient distributed solution [17]. To be precise, the central peak-point probabilities of the PA contained in a cluster of the local optimal solutions are found by implementing a distributed search algorithm via message-passing techniques [20, 21]. Therefore, individual users may be able to participate in obtaining computationally demanding solutions. A detailed description of the strategy is presented in the following sections.

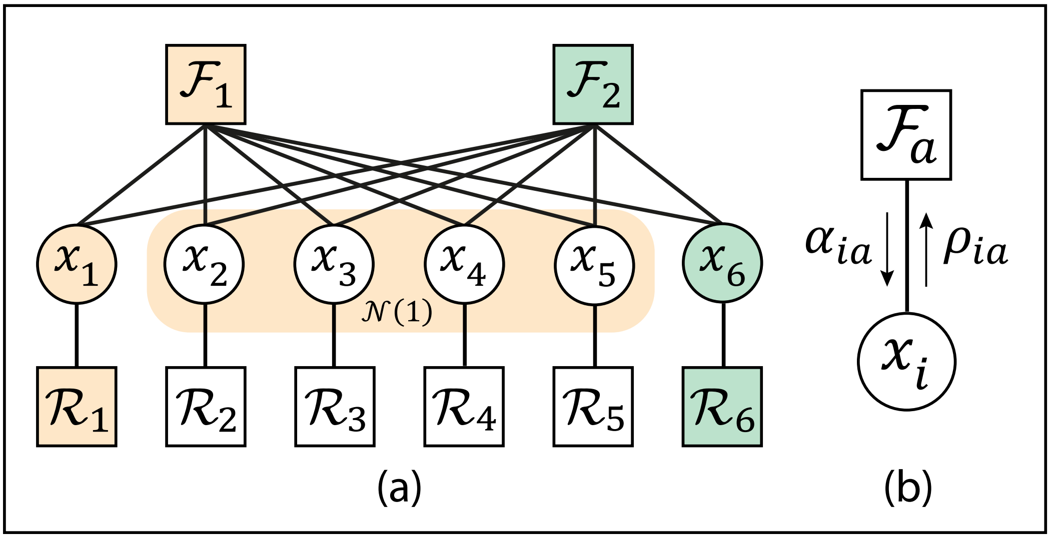


Figure 4.1: Graphical representation for pilot assignment problem. (a) Factor graph model for **(P2)**. (b) Two types of messages between $\mathcal{F}_a$ and x_i.

4.1 Graphical Representation

To derive the corresponding message-passing based distributed algorithm, the structure of a graphical model that facilitates the understanding of the problem structure is necessarily considered. Fig. 4.1(a) illustrates a factor graph model [20] associated with the toy example. Note that if the group $\mathcal{K}_1$ is used for assignment, it limits the groups in $\mathcal{N}(1)$ unused. While the use of $\mathcal{K}_1$ constrains the states of corresponding neighbor groups, other states are not determined but are prohibited from assigning the same pilot with $\mathcal{K}_1$.

4.2 Derivation of Algorithm

4.2.1 Design Principles for Computation

According to generic message-passing computational rules [20], the tendency for the allocation of the a-th pilot to group $\mathcal{K}_i$ in optimal states are characterized by messages exchanged between node x_i and node $\mathcal{F}_a$. To this end, two different types of messages are defined, respectively, as shown in Fig. 4.1(b): $\rho_{ia}(x_i = x)$, a message transferred from x_i to $\mathcal{F}_a$ informing how evident the value of x_i can take on a value of x, and $\alpha_{ia}(x_i = x)$, a message sent from $\mathcal{F}_a$ to x_i representing how preferable the value of $x_i = x$ is to satisfy the constraint enforced by the factor function $\mathcal{F}_a$. Under equilibrium states, both messages necessarily represent certain relationships between group $\mathcal{K}_i$ and a-th pilot is given according to the state of x_i to maximize unconstrained formulation (**P2**) as

$$\rho_{ia}(x_i) = \mathcal{R}_i(x_i) + \sum_{b \neq a} \alpha_{ib}(x_i)\,, \tag{4.1}$$

$$\alpha_{ia}(x_i) = \max_{\mathbf{x} \setminus \{x_i\}} \left(\mathcal{F}_a(\mathbf{x}) + \sum_{j \neq i} \rho_{ja}(x_j) \right). \tag{4.2}$$

Thus, $\rho_{ia}(x_i)$ and $\alpha_{ia}(x_i)$ take the values associated with the maximal objective when group $\mathcal{K}_i$ is assigned to the a-th pilot.

However, there may exist too many states of a PA task that slightly differ from optimal state $\mathbf{x}$ and take corresponding objective values very close to the optimum. In order to control all these assignments collectively, a basic principle of handling combinatorial optimization with the SP framework is applied by introducing multiple possible states for individual discrete variables, representing *chosen*, *unchosen*, and *undetermined*. In the PA task, the state of an individual variable x_i imposed by the constraint $\mathcal{F}_a$ can be categorized into three types: *assigned*, *idle*, and *else-*

where. If $x_i = a$, any group that shares members with $\mathcal{K}_i$ becomes idle, and no other group is assigned to the a-th pilot. If $x_i = 0$, which means that $\mathcal{K}_i$ is idle, other groups do not care about it and go into the so-called don't-care state[16]. If $x_i \notin \{0, a\}$, which is associated with the elsewhere state, only the neighbor groups of $\mathcal{K}_i$ are restricted to being idle.

To consolidate the relationship among these three states, three different types of messages are defined, i.e., $\{\rho_{ia}^{(a)}, \alpha_{ia}^{(a)}\}$ and $\{\rho_{ia}^{(0)}, \alpha_{ia}^{(0)}\}$ along with new messages associated with elsewhere state, $\{\rho_{ia}^{(*)}, \alpha_{ia}^{(*)}\}$.

4.2.2 Messages for Assigned States

If $x_i = a$, the message $\rho_{ia}(x_i)$ is computed as follows.

$$\rho_{ia}(a) = \mathcal{R}_i(a) + \sum_{b \neq a} \alpha_{ib}(a) = R_{ia} + \sum_{b \neq a} \alpha_{ib}(a)\,. \tag{4.3}$$

Since the group $\mathcal{K}_i$ is assigned elsewhere from the b-th pilot such that $b \neq a$,

$$\rho_{ia}^{(a)} = R_{ia} + \sum_{b \neq a} \alpha_{ib}^{(*)}\,. \tag{4.4}$$

In the same case, the message $\alpha_{ia}(x_i)$ is computed as follows.

$$\alpha_{ia}(a) = \max_{\mathbf{x} \setminus \{x_i\}} \left(\mathcal{F}_a(\mathbf{x}) + \sum_{j \neq i} \rho_{ja}(x_j) \right). \tag{4.5}$$

To avoid the penalizing function $\mathcal{F}_a$ returns negative infinity, all of the other state variables in $\mathbf{x}$ are prohibited to be a. The assigned message $\alpha_{ia}^{(a)}$ is then given by

$$\alpha_{ia}^{(a)} = \sum_{i' \in \mathcal{N}(i)} \rho_{i'r}^{(0)} + \sum_{i'' \in \mathcal{N}(i)^c \setminus \{i\}} \max\left(\rho_{i''r}^{(0)}, \rho_{i''r}^{(*)} \right). \tag{4.6}$$

4.2.3 Messages for Idle States

If $x_i = 0$, the message $\rho_{ia}(x_i)$ is computed as follows.

$$\rho_{ia}(0) = \mathcal{R}_i(0) + \sum_{b \neq a} \alpha_{ib}(0) = \sum_{b \neq a} \alpha_{ib}(0)\,. \tag{4.7}$$

The corresponding idle message is straightforwardly given as follows, although the message $\alpha_{ia}^{(0)}$ is not.

$$\rho_{ia}^{(0)} = \sum_{b \neq a} \alpha_{ib}^{(0)}\,. \tag{4.8}$$

To compute the idle message $\alpha_{ia}^{(0)}$, we divide the idle state into two cases.

- **a-th pilot is utilized anyway:**
  There is only one $j \neq i$ s.t. $x_j = a$,
  then $x_{j'} = 0$ for $j' \in (N(j) \backslash \{i\})$ and $x_{j''} \neq a$ for $j'' \notin (N(j) \cup \{i\})$.

- **a-th pilot is not used for the uplink training:**
  $\nexists j \neq i$ s.t. $x_j = a$.

Since the message $\alpha_{ia}^{(0)}$ returns the maximum value among these cases, it is given by

$$\alpha_{ia}^{(0)} = \max \left\{ \begin{array}{l} \max\limits_{j \neq i} \left(\rho_{ja}^{(a)} + \sum\limits_{j' \in \mathcal{N}(j) \backslash \{i\}} \rho_{j'a}^{(0)} + \sum\limits_{j'' \in \mathcal{N}(j)^c \backslash \{i,j\}} \max\left(\rho_{j''r}^{(0)},\ \rho_{j''r}^{(*)} \right) \right), \\ \\ \sum\limits_{j \neq i} \max\left(\rho_{ja}^{(0)}, \rho_{ja}^{(*)} \right). \end{array} \right. \tag{4.9}$$

4.2.4 Messages for Elsewhere States

For representing a number of elsewhere states, we pick the maximum value of $\rho_{ia}(x_i)$ and $\alpha_{ia}(x_i)$, while $x_i \notin \{0, a\}$, respectively. The message $\rho_{ia}^{(*)}$ is then given by

$$\rho_{ia}^{(*)} = \max_{b\neq a}\left(R_{ib} + \alpha_{ib}^{(b)} + \sum_{s\notin\{a,b\}} \alpha_{is}^{(*)}\right). \tag{4.10}$$

We divide the idle state into two cases as same as $\alpha_{ia}^{(0)}$ to compute the idle message $\alpha_{ia}^{(*)}$, but the states of neighbor groups are limited to be idle in both of cases. Remainer descriptions of each case are as follows.

- **a-th pilot is utilized anyway:**
  There is only one $j \notin (\mathcal{N}(i) \cup \{i\})$ s.t. $x_j = a$,
  then $x_{j'} = 0$ for $j' \in (N(i) \cup N(j))$ and $x_{j''} \neq a$ for others.

- **a-th pilot is not used for the uplink training:**
  There is no state variable x_j such that $x_j = a$.

Note that if $j \notin \mathcal{N}(i)$, $i \notin \mathcal{N}(j)$. The corresponding message is given by

$$\alpha_{ia}^{(*)} = \sum_{i'\in\mathcal{N}(i)} \rho_{i'a}^{(0)} + \max\begin{cases} \max\limits_{j\in\mathcal{N}(i)^c\setminus\{i\}} \left(\rho_{ja}^{(a)} + \sum\limits_{j'\in\mathcal{N}(j)\setminus\mathcal{N}(i)} \rho_{j'a}^{(0)} + \sum\limits_{j''\in\mathcal{N}(j)^c\setminus(\{i\}\cup\mathcal{N}(i))} \max\left(\rho_{j''a}^{(0)}, \rho_{j''a}^{(*)}\right)\right), \\ \sum\limits_{i''\in\mathcal{N}(i)^c\setminus\{i,j\}} \max\left(\rho_{i''a}^{(0)},\ \rho_{i''a}^{(*)}\right). \end{cases} \tag{4.11}$$

4.3 Rules for Message Update and Decision

The message-passing algorithm is conducted to collect the information of optimality that is encoded as aforementioned descriptions. Upon convergence of the messages, The decision of the value for x_i is made using

$$\tilde{b}_{ia} \triangleq (\rho_{ia}^{(a)} - \rho_{ia}^{(0)}) + (\alpha_{ia}^{(a)} - \alpha_{ia}^{(0)})\,, \tag{4.12}$$

$$\bar{b}_{ia} \triangleq (\rho_{ia}^{(a)} - \rho_{ia}^{(*)}) + (\alpha_{ia}^{(a)} - \alpha_{ia}^{(*)})\,. \tag{4.13}$$

If group index a is found with possible $\max_a \tilde{b}_{ia}$, the i-th users combination is assigned the $\hat{x}_i$-th pilot resource, which is given by $\hat{x}_i = \arg\max_a \bar{b}_{ia}$. Otherwise, the i-th users combination is released idle.

Note that the finally used forms of the messages are only two measures, difference between assigned state and idle state ($\{\rho_{ia}^{(a)} - \rho_{ia}^{(0)},\ \alpha_{ia}^{(a)} - \alpha_{ia}^{(0)}\}$), difference between assigned state and elsewhere state ($\{\rho_{ia}^{(a)} - \rho_{ia}^{(*)},\ \alpha_{ia}^{(a)} - \alpha_{ia}^{(*)}\}$). While the compactness of the message exchange is essential, it is sufficient to make decisions by the compressed forms of messages that are reduced from three to two types. The corresponding measures can be defined and calculated, respectively, as

$$\tilde{\rho}_{ia} \triangleq \rho_{ia}^{(0)} - \rho_{ia}^{(*)} = \sum_{b \neq a} (\tilde{\alpha}_{ib} - \bar{\alpha}_{ib}) - \max_{b \neq a} (R_{ib} - \bar{\alpha}_{ib})\,, \tag{4.14}$$

$$\bar{\rho}_{ia} \triangleq \rho_{ia}^{(a)} - \rho_{ia}^{(*)} = R_{ia} - \max_{b \neq a} (R_{ib} - \bar{\alpha}_{ib})\,, \tag{4.15}$$

$$\begin{aligned} \tilde{\alpha}_{ia} &\triangleq \alpha_{ia}^{(0)} - \alpha_{ia}^{(a)} \\ &= \left[\max_{j \neq i} \left(\bar{\rho}_{ja} - [\tilde{\rho}_{ja}]^+ + \sum_{j' \in \mathcal{N}(j) \setminus \{i\}} [\tilde{\rho}_{j'a}]^- \right) \right]^+ - \sum_{i' \in \mathcal{N}(i)} [\tilde{\rho}_{i'a}]^-\,, \end{aligned} \tag{4.16}$$

$$\begin{aligned} \bar{\alpha}_{ia} &\triangleq \alpha_{ia}^{(*)} - \alpha_{ia}^{(a)} \\ &= \left[\max_{j \in \mathcal{N}(i)^c \setminus \{i\}} \left(\bar{\rho}_{ja} - [\tilde{\rho}_{ja}]^+ + \sum_{j' \in \mathcal{N}(j) \setminus \mathcal{N}(i)} [\tilde{\rho}_{j'a}]^- \right) \right]^+, \end{aligned} \tag{4.17}$$

where $[\ \cdot\ ]^+ = \max(0,\ \cdot\ )$, $[\ \cdot\ ]^- = \min(0,\ \cdot\ )$. Then the decision variables are caculated as followings.

Algorithm 1 Distributed PA algorithm

Input: Partial sum-rate $\{R_{ia}\}$ for all (i, a) in (3.1).

Output: Pilot indices of all users $\{x_i\}$

1: *Initialization* : Set $\tilde{\alpha}_{ia}^{[1]} = 0$, and $\bar{\alpha}_{ia}^{[1]} = 0$ for all (i, r) and $t \leftarrow 1$.

repeat

Message updates

2: Update $\tilde{\rho}_{ia}^{[t]}$ and $\bar{\rho}_{ia}^{[t]}$ using (4.14) and (4.15), respectively.

3: Update $\tilde{\alpha}_{ia}^{[t+1]}$ and $\bar{\alpha}_{ia}^{[t+1]}$ using (4.16) and (4.17).

Tentative decision

4: Evaluate $\tilde{b}_{ia}$ and $\bar{b}_{ia}$ using (4.18) and (4.19).

5: **if** $(\max_a \tilde{b}_{ia} > 0)$ **then** $x_i \leftarrow \arg\max_a \bar{b}_{ia}$.

6: **else** $x_i \leftarrow 0$.

7: *Increment* : $t \leftarrow t + 1$.

until Convergence of all messages

8: **return** Allocate pilot according to $\{x_i\}$.

$$\tilde{b}_{ia} = (\bar{\rho}_{ia} - \tilde{\rho}_{ia}) - \tilde{\alpha}_{ia}\,, \tag{4.18}$$

$$\bar{b}_{ia} = \bar{\rho}_{ia} - \bar{\alpha}_{ia}\,. \tag{4.19}$$

Algorithm 1 summarizes the overall SP-based distributed algorithm, where individual message update computations can be handled by individual users and tentative decisions are made by the CPU interconnected to APs.

Chapter 5

Numerical Results

5.1 Experimental Setup

This section presents numerical results to test the performance of the developed algorithm. A CF massive MIMO configuration is considered so that M APs and K users are equipped with a single antenna and randomly distributed over a 500m $\times$ 500m square area. The simulation region is wrapped around in order to mitigate the boundary effects [1]. The heights of APs and users are set as 10m and 1.5m, respectively. The communication bandwidth of 20MHz and carrier frequency of 2GHz are used. The normalized transmit SNR is set to $\omega = \bar{\omega}/\omega_{\text{noise}}$, with $\bar{\omega} =$ 100mW and the noise power $\omega_{\text{noise}} = -92$dBm. The 3GPP Urban Microcell path loss model is used to set large-scale propagation conditions, where the shadow fading model is set to $z_{mk} \sim \mathcal{N}(1, \sigma_{\text{sh}}^2)$ with $\sigma_{\text{sh}} = 4$dB [22, Table B.1.2.1-1]. The length of the coherence interval N_{c} is taken as 200. We assume that the channels are constant during the coherence interval, and all users transmit with full power so that $\eta_k = 1$.

The proposed algorithm is compared with three existing PA schemes based on greedy algorithm [1], K-means clustering [4], and Hungarian algorithm [9]. The greedy strategy repetitively identifies the pilot-user pair with the least user throughput and assigns a new pilot sequence to the identified user while fixing all other PA configurations. In K-means clustering based PA, all users are grouped by N_{p} members, then pilots are assigned orthogonally to users in each cluster. At this time, $\lceil K/N_{\mathrm{p}} \rceil$ centroids for the deployment of the APs are used for user clustering, and the centroids have been trained through the K-means clustering algorithm. On the other hand, Hungarian algorithm-based strategy execute the following tasks sequentially for each user: select $N_{\mathrm{p}} - 1$ users with the highest proximity and conduct the bipartite matching for PA while the assignment of other users is fixed. In each simulation, N_{p} users are chosen at random and orthogonal pilots are preassigned to them so that PA is needed for the other $K - N_{\mathrm{p}}$ users. Note that pre-assignment for some users poses a risk of performance degradation instead of improving the PA complexity. For other benchmarks, we set up those to show full performance without pre-assignment. We evaluate the additional gain for SP framework with respect to Hungarian algorithm, K-means clustering, and greedy algorithm, respectively. The corresponding percentages are calculated based on the performance of each benchmark.

5.2 Results and Discussions

Fig. 5.1 illustrates the cumulative distribution function of the system throughput in uplink with $M = 100$, $K = 45$, and $N_{\mathrm{p}} = 15$. The proposed algorithm, SP, outperforms other benchmarks constantly over the simulation, thus indicating that the pilot contamination is effectively mitigated. The average system throughput obtained by SP is about 744 Mbps, which corresponds to the improvement over compared schemes by 2%, 5%, and 10.3% respectively. In addition, a popular metric for PA is the network throughput for the worst 5% PA configurations [4, 9]. SP obtains the value of 657 Mbps, and this amounts to 3%, 6%, and 13% improvement over others, respectively. Thus, the population of low throughput

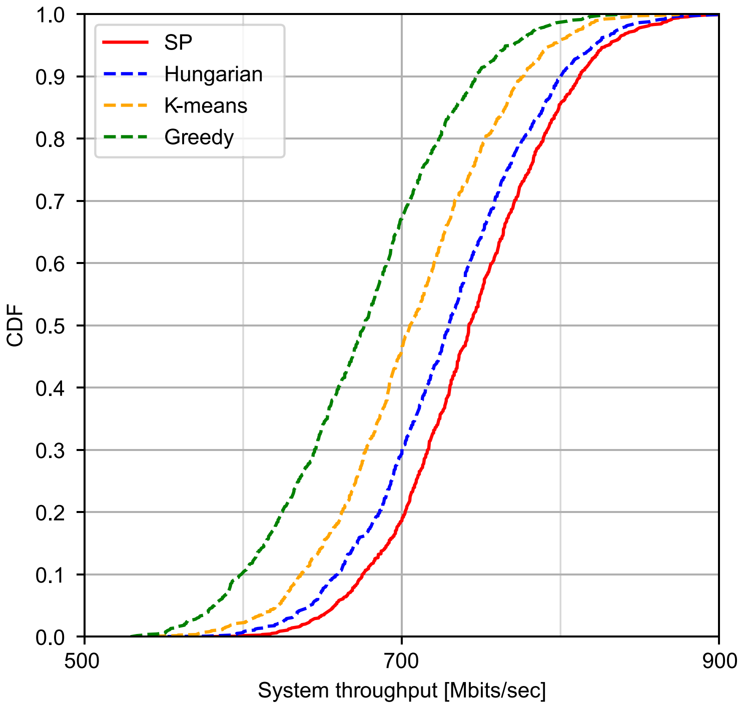


Figure 5.1: Cumulative distribution function of system throughput where $K = 45, M = 100, N_{\mathrm{p}} = 15$.

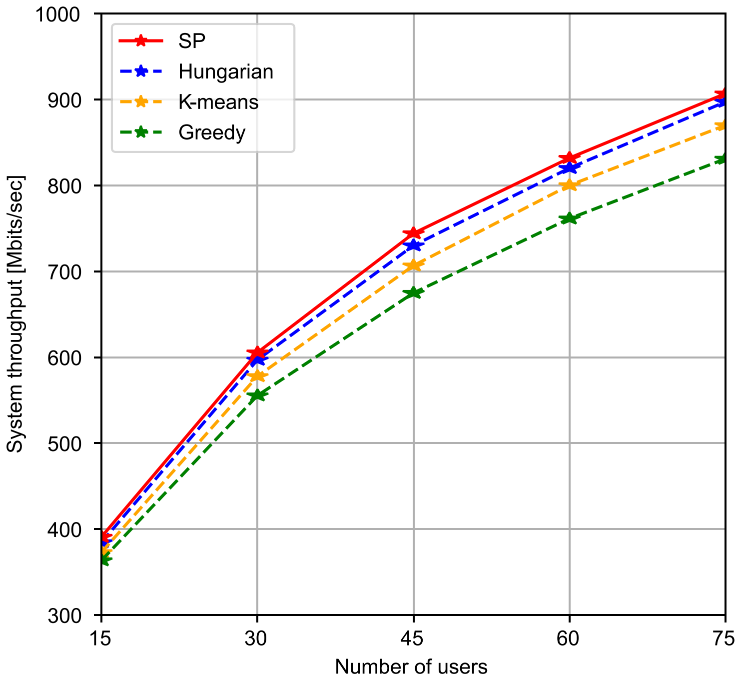


Figure 5.2: Average system throughput versus number of users where $K \in \{15, 30, 45, 60, 75\}, M = 100, N_{\rm p} = 15$.

users is decreased by the proposed method.

Fig. 5.2 depicts the average system throughput where $M = 100$ and $K \in \{15, 30, 45, 60, 75\}$. The number of users per orthogonal pilot resources, $K/N_{\rm p}$, is fixed to 3. For large user populations, the system throughput growth under all PA strategies attenuates due to the additional beamforming uncertainty and user interference from the network congestion. At the largest population of $K = 75$, the SP framework results about 907 Mbps then outperforms Hungarian algorithm, K-means clustering, and greedy strategies, respectively, by 1%, 4%, and 9%. This shows the robustness to the network congestion with respect to benchmarks.

The SP approach is suited to optimize corresponding tasks for network throughput maximization from the advantage achieved by organizing the PA task from a group-wise matching framework. This allows the achievable rate to be directly utilized in network management. Through the developed algorithm, the PA operation can be configured to explore the solution space of the PA efficiently.

Chapter 6

Conclusion

This work develops a survey propagation based distributed algorithm that conducts the CF massive MIMO pilot assignment optimization. The developed algorithm can efficiently describe the association of groups of multiple users to a pilot sequence to be shared and determine the corresponding best matching implemented via a message-passing approach. Numerical experiments demonstrate that the proposed framework outperforms other existing schemes with respect to the overall network throughput.

Bibliography

[1] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wirel. Commun.*, vol. 16, pp. 1834–1850, March 2017.

[2] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, "Prospective multiple antenna technologies for beyond 5G," *IEEE J. Sel. Areas Commun.*, vol. 38, pp. 1637–1660, Aug. 2020.

[3] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions," *IEEE Commun. Surveys & Tut.*, vol. 24, pp. 611–652, Jan. 2022.

[4] S. Chen, J. Zhang, E. Björnson, J. Zhang, and B. Ai, "Structured massive access for scalable cell-free massive MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 39, pp. 1086–1100, April 2021.

[5] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 16, pp. 4445–4459, July 2017.

[6] E. Björnson *et al.*, *Massive MIMO networks: Spectral, energy, and hardware efficiency*, vol. 11. Now Publishers, Inc., 2017.

[7] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wirel. Commun. Net.*, vol. 2019, no. 1, pp. 1–13, 2019.

[8] H. Yu, X. Yi, and G. Caire, "Topological pilot assignment in large-scale distributed MIMO networks," *IEEE Trans. Wirel. Commun.*, vol. Early Access, pp. 1–1, 2022.

[9] S. Buzzi, C. D'Andrea, M. Fresia, Y.-P. Zhang, and S. Feng, "Pilot assignment in cell-free massive MIMO based on the hungarian algorithm," *IEEE Wirel. Commun. Lett.*, vol. 10, pp. 34–37, Jan. 2021.

[10] W. Zeng, Y. He, B. Li, and S. Wang, "Pilot assignment for cell free massive MIMO systems using a weighted graphic framework," *IEEE Trans. Veh. Tech.*, vol. 70, pp. 6190–6194, June 2021.

[11] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "Downlink resource allocation in multiuser cell-free mimo networks with user-centric clustering," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1482–1497, 2022.

[12] H. Liu, J. Zhang, X. Zhang, A. Kurniawan, T. Juhana, and B. Ai, "Tabu-search-based pilot assignment for cell-free massive mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2286–2290, 2020.

[13] H. Liu, J. Zhang, S. Jin, and B. Ai, "Graph coloring based pilot assignment for cell-free massive mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9180–9184, 2020.

[14] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: journal of the Econometric Society*, pp. 53–76, 1957.

[15] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the ACM (JACM)*, vol. 23, no. 3, pp. 555–565, 1976.

[16] A. Braunstein, M. Mézard, and R. Zecchina, "Survey propagation: An algorithm for satisfiability," *Random Structures & Algorithms*, vol. 27, no. 2, pp. 201–226, 2005.

[17] C. Srinivasa, S. Ravanbakhsh, and B. Frey, "Survey propagation beyond constraint satisfaction problems," in *Artificial Intelligence and Statistics*, pp. 286–295, 2016.

[18] J. H. Sørensen, E. de Carvalho, Stefanovic, and P. Popovski, "Coded pilot random access for massive MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 17, pp. 8035–8046, Dec. 2018.

[19] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, vol. 9. World Scientific Publishing Company, 1987.

[20] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[21] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[22] 3GPP, "Further advancements for E-UTRA physical layer aspects (Release 9)," Tech. Rep. 36.814 v9.2.0, March 2017.

Appendix A

Python Codes

Our demonstration has been conducted in a python environment. The following python codes execute the message updates and decision process. Note that this is abstract information for the implementation of SP algorithm and other functions such as weight(partial sum rate) generator are required for the simulation.

```
import numpy as np


# x: state vector, x_j0: state of preallocated user


def configurations(dim_x, n_pilot, neighbor_mapping, x_j0):
    map_neighbor = np.zeros((dim_x, dim_x), dtype=int)
    j0_prime = np.zeros((dim_x, dim_x, dim_x), dtype=int)
    j0_window = np.zeros((dim_x, dim_x, n_pilot), dtype=int)
    for i in range(dim_x):
        neighbor_mapping[i, neighbor_mapping[i]] = 1
        j0_window[i, x_j0[i], :] = 1
        for j0 in x_j0[i]:
            j0_prime = list(set(neighbor_mapping[j0]) - set(
```

```
            ↪ neighbor_mapping[i]))
        j0_prime[i, j0, j0_prime] = 1
    j_prime = np.tile(neighbor_mapping, (dim_x, 1, 1))
    col_window = 1 - np.tile(np.expand_dims(np.eye(dim_x, dtype=int)
        ↪ , axis=1), (1, dim_x, 1))
    j_prime[col_window==0] = 0
    row_window = 1 - np.tile(np.expand_dims(np.eye(dim_x), axis=2),
        ↪ n_pilot)
    return neighbor_mapping, j_prime, j0_prime, row_window,
        ↪ j0_window

# R: matrix of partial sum-rates
# rT: rT, rB: rB
# aT: aT, aB: aB

def update_rho(R, aT_now, aB_now):
    dim_x, n_pilot = np.shape(R)
    rB_now = np.zeros(shape=(dim_x, n_pilot))
    rT_now = np.zeros(shape=(dim_x, n_pilot))
    for r in range(n_pilot):
        aT_except_r = np.delete(aT_now, r, axis=1)
        aB_except_r = np.delete(aB_now, r, axis=1)
        R_except_r = np.delete(R, r, axis=1)
        rT_now[:, r] = R[:, r] + np.sum(
            aT_except_r - aB_except_r, axis=1)
        rB_now[:, r] = R[:, r] - np.max(
            R_except_r + aB_except_r, axis=1)
    return rT_now, rB_now
```

```
def update_alpha(map_neighbor, j_prime, j0_prime,
                 row_window, j0_window, rT_now, rB_now):
    dim_x, _ = np.shape(rT_now)
    diff_rBT_n = np.minimum(rB_now - rT_now, 0)
    diff_rBT_p = np.maximum(rB_now - rT_now, 0)
    rB_tile = np.tile(rB_now, (dim_x, 1, 1))
    diff_rBT_p_tile = np.tile(diff_rBT_p, (dim_x, 1, 1))
    diff_rBT_n_tile = np.tile(diff_rBT_n, (dim_x, 1, 1))
    j_prime_term_tile = np.matmul(j_prime, diff_rBT_n_tile)
    j0_prime_term_tile = np.matmul(j0_prime, diff_rBT_n_tile)

    term1 = np.matmul(map_neighbor, diff_rBT_n)
    term2 = -rB_tile + diff_rBT_p_tile - j_prime_term_tile
    term2[row_window==0] = INFIN
    aT_next = term1 + np.minimum(np.min(term2, axis=1), 0)

    term3 = -rB_tile + diff_rBT_p_tile - j0_prime_term_tile
    term3[j0_window==0] = INFIN
    aB_next = np.minimum(np.min(term3, axis=1), 0)

    return aT_next, aB_next



def make_decision(x, aT_now, aB_now,
                  rT_now, rB_now):
    dim_x = len(x)
    dim_r = np.size(aT_now, axis=1)
    allocation = np.zeros(dim_x)
```

```
b_tilde = aT_now + rT_now
b_bar = aB_now + rB_now
for i in range(dim_x):
    if np.max(b_tilde[i, :]) > 0:
        allocation[i] = np.argmax(b_bar[i, :])
    else:
        allocation[i] = None
for r in range(dim_r):
    if np.count_nonzero(allocation==r) == 0:
        i_argmax = np.argmax(b_tilde[:, r])
        allocation[i_argmax] = r
return allocation
```