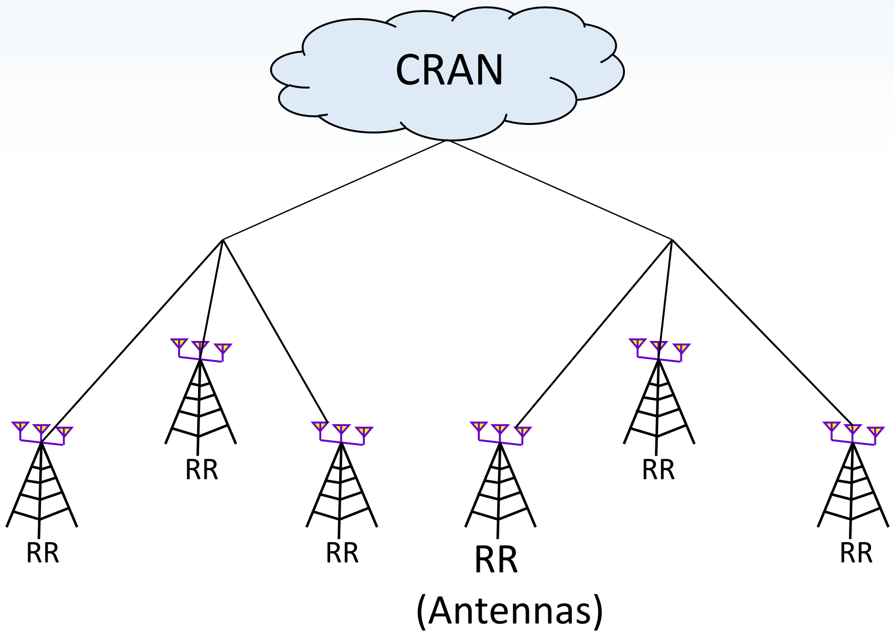
“Wireless Networks Without Edges”: Dynamic Radio Resource Clustering and User Scheduling

Yuhuan Du, Gustavo de Veciana

(*University of Texas at Austin*)

Cloud-based Radio Access Networks (CRANs)



- RRs distributed to serve users.
- Computing tasks done at computing center.

Benefits:

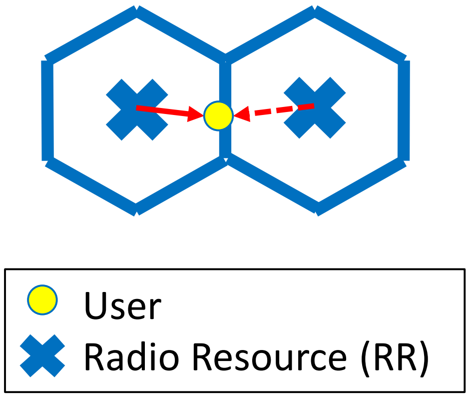
- Reduced setup and maintenance cost
  - Resource sharing
  - Fewer centers
  - Simple RRs
- Facilitate cooperation

Challenges:

- Tight turnaround deadline (in LTE, deadline is 1-10ms)

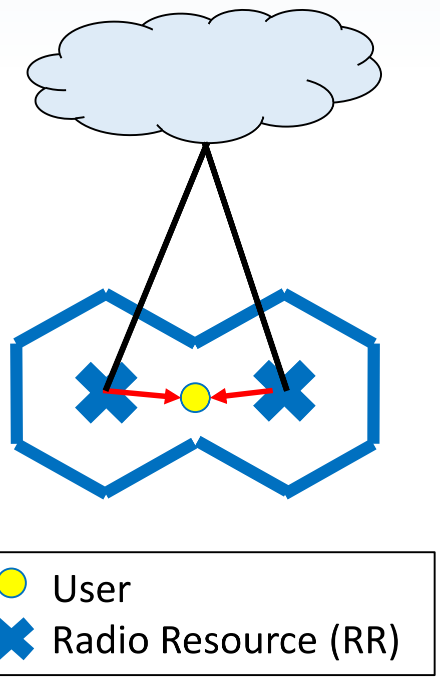
Cooperation in CRAN

Consider Downlink



Cooperation in CRAN

Consider Downlink



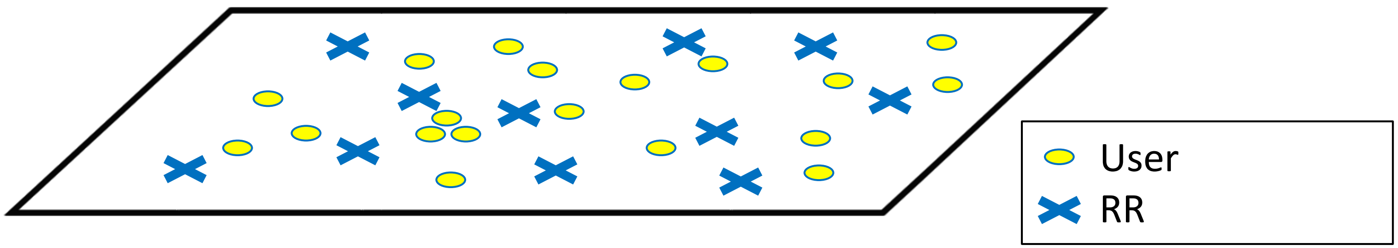
- Coordinated Multi-Point (CoMP) Tx/Rx helps to control interference
- Why not cooperate across all RRs? *
- Clustering RRs into static collections is not enough

We need to dynamically cluster RRs into collections of limited size.

*A. Lozano, R. W. H. Jr., and J. G. Andrews, “Fundamental limits of cooperation”.

Problem Formulation:
Utility-Driven Resource Allocation

Downlink, best-effort traffic, OFDMA-based system

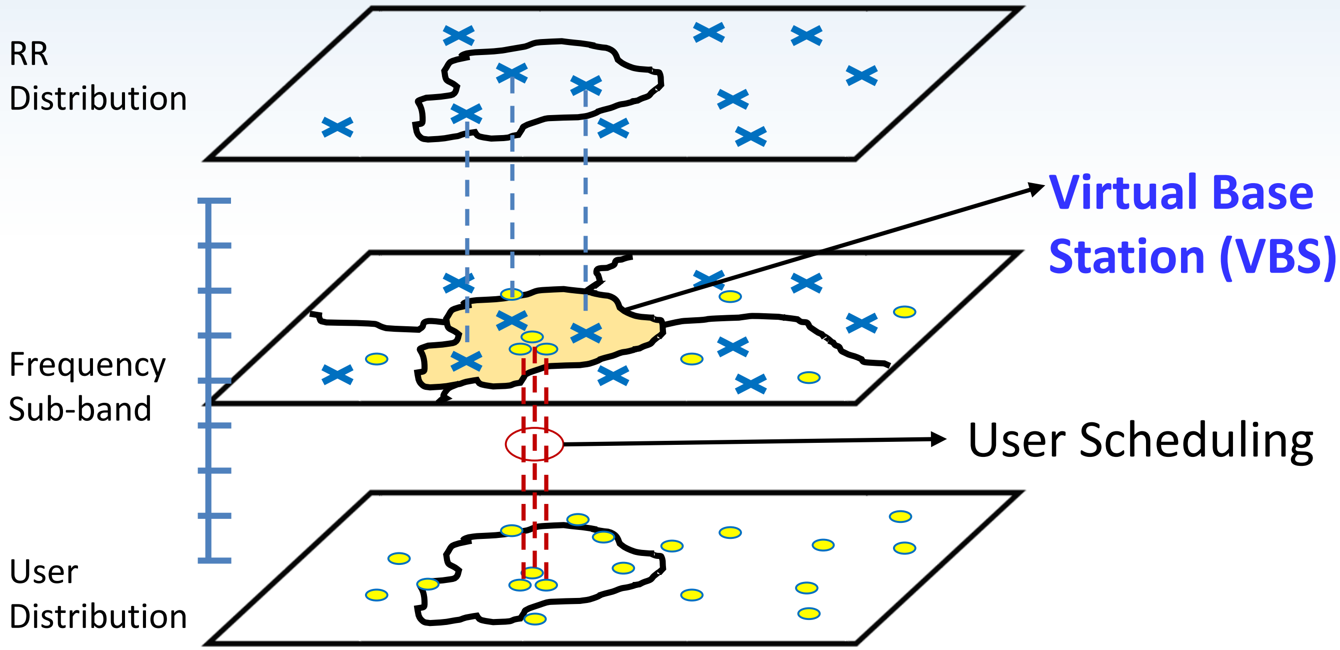


$$\max_{\substack{\text{resource decisions} \\ \text{over time \& freq.}}} \sum_i U_i(\boxed{X_i})$$

average rate

concave

What Needs to Be Decided?



For each time & sub-band need to choose

- VBS partition (RR clusters) and schedule associated users
- Tx power & precoding vectors, etc.

Choices are interdependent because of interference

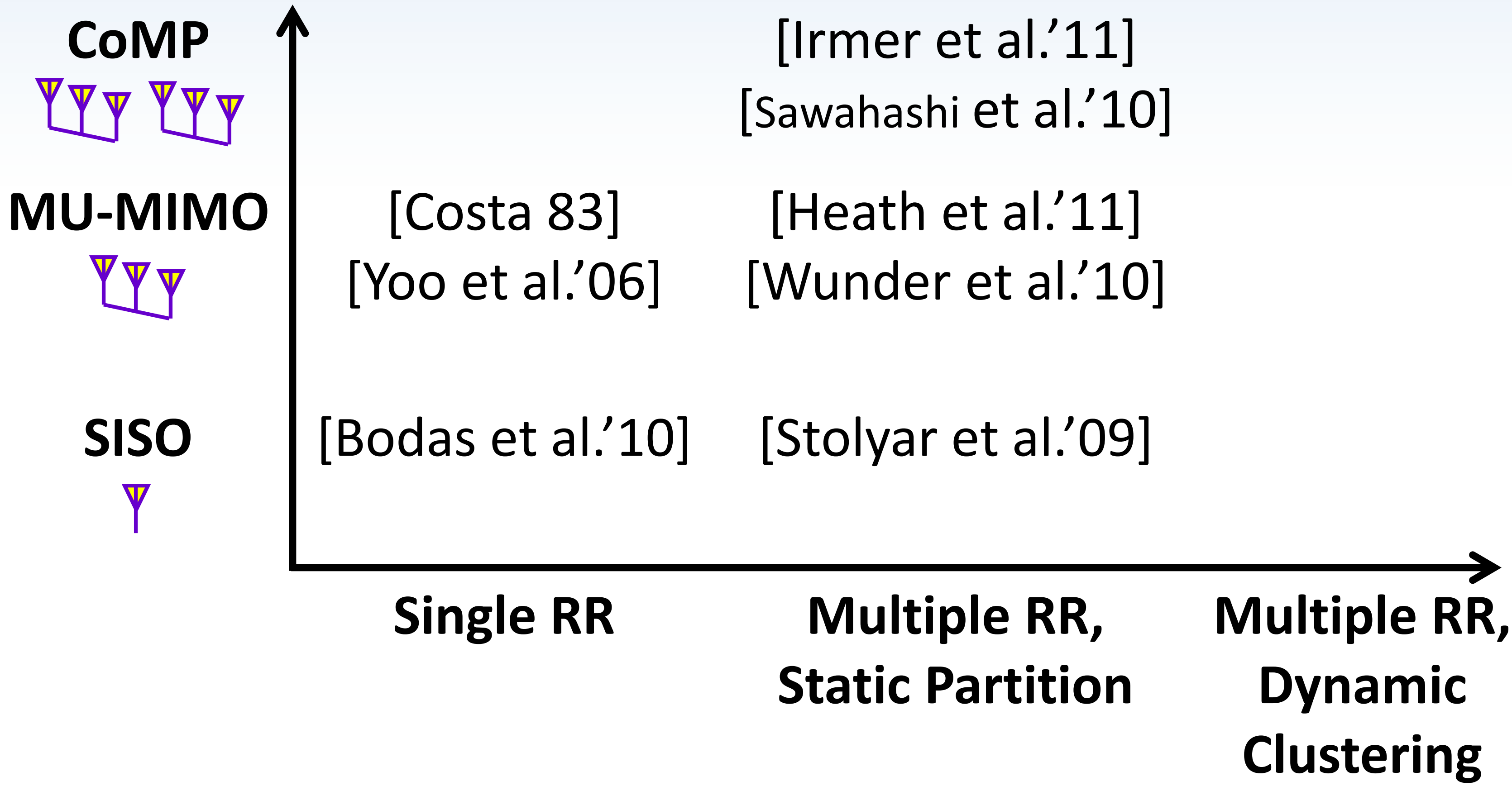
This Problem is Extremely “Complex”

- Need to be solved over and over again.

- We do not aim to perfectly solve this problem.

- Instead, we aim to adopt several suboptimal techniques to decompose this problem for efficient computation.

Related Work



Recall Problem

$$\max_{\text{resource decisions over time \& freq.}} \sum_i U_i(\boxed{X_i})$$

average rate

Theorem*: If $U_i(\cdot)$ are concave, this can be solved asymptotically by using the following greedy strategy at each time & sub-band

$$\max_{\text{resource decision } d} \sum_i$$ Marginal Utility for user i under decision d

*A. L. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation".

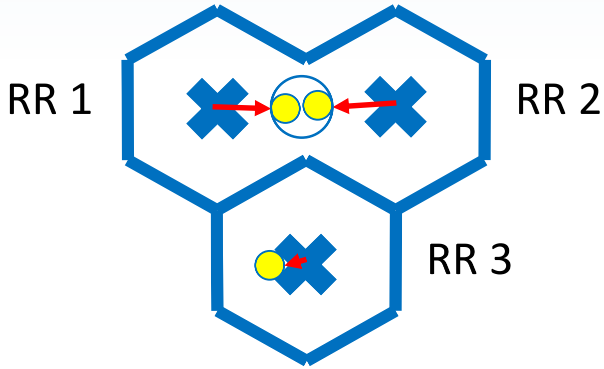
Approach: Dealing with Interdependence

Slowly adapt RRs' power allocation limits $\mathbf{P}$ on different **sub-bands***

- Pros
  - Make "interference" from other VBSs predictable
    - Decouple decisions across VBSs
  - Combined with CoMP to provide instantaneous power allocation to users given constraints $\mathbf{P}$
    - Explore opportunistic benefits
- Cons
  - Lose some flexibility, but not much

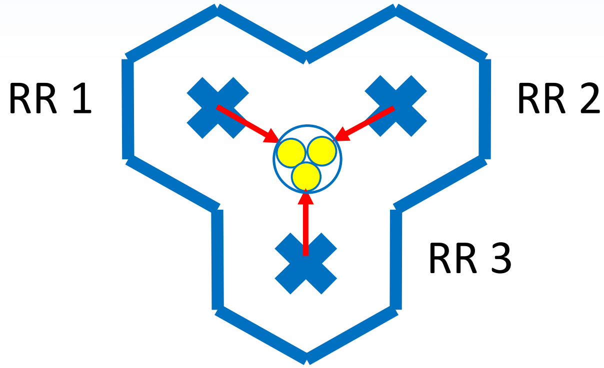
*Based on ideas used in prior work by Stolyar et al.

$\mathbf{P} \longleftrightarrow$ Desirable Partitioning & Scheduling



Power allocation table:

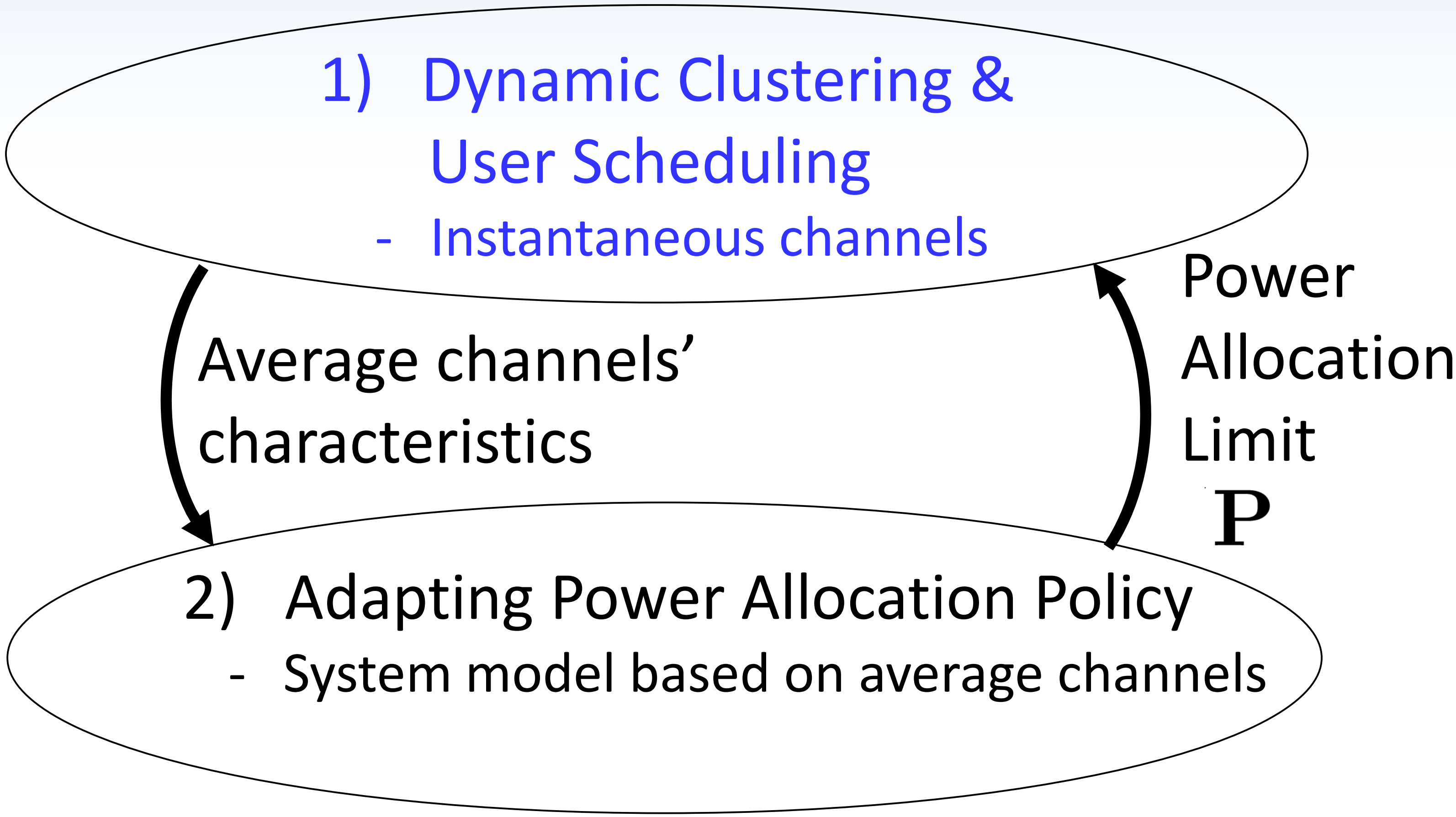
	RR 1	RR 2	RR 3
…	…	…	…
Sub-band j	High	High	Low



Power allocation table:

	RR 1	RR 2	RR 3
…	…	…	…
Sub-band j	High	High	Low

Framework


1) Dynamic Clustering & User Scheduling
- Instantaneous channels
Average channels' characteristics
Power Allocation Limit
$\mathbf{P}$
2) Adapting Power Allocation Policy
- System model based on average channels

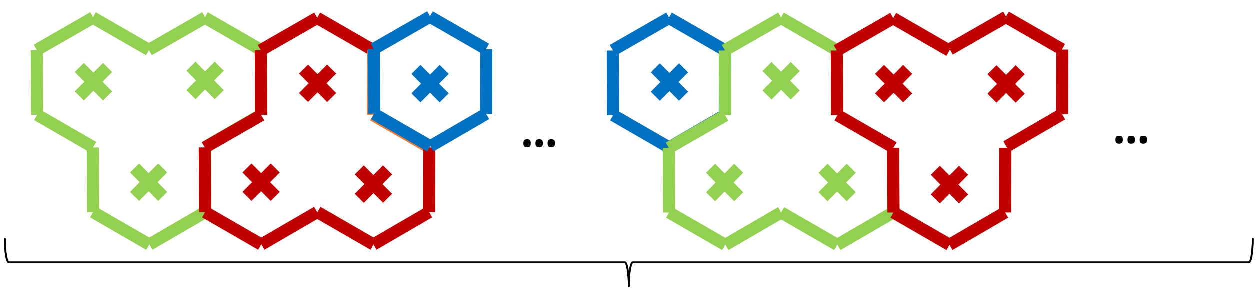
Dynamic Clustering & User Scheduling

Inputs:
- Utility fn
- Channels
- Avg rates
- $\mathbf{P}$ / RR & sub-band

Ouputs:
- VBS partition
- User scheduling
- Power per user

Phy. Layer:
- Precoder

How do we choose optimal VBS partitions?



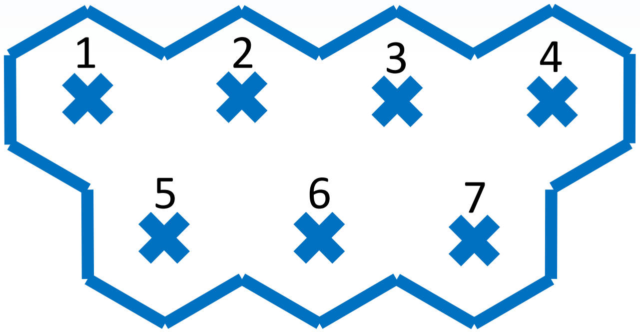
Exponential number of possible partitions

- Assign weight (measure of “marginal utility”) to each VBS
- Find Max Weight Exact Cover $\longleftrightarrow$ NP-hard

Our Solution: Structuring Set of Possible VBSs

Theorem: If the set of possible VBSs is **2-decomposable**, the maximum weight exact cover problem is solvable in polynomial time.

E.g.



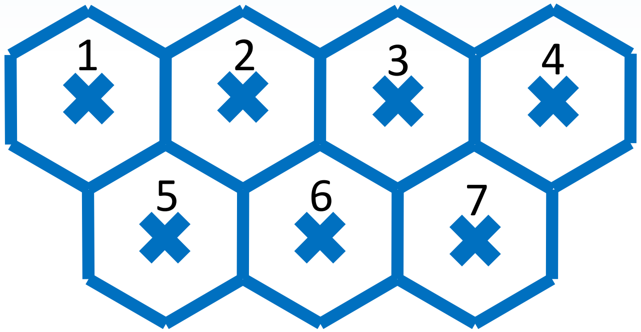
Possible VBSs:

{1}, {2}, {3}, {4},
{5}, {6}, {7},
{1,2,5}, {3,6,7},
{2,5,6}, {3,4,7}

Our Solution: Structuring Set of Possible VBSs

Theorem: If the set of possible VBSs is **2-decomposable**, the maximum weight exact cover problem is solvable in polynomial time.

E.g.



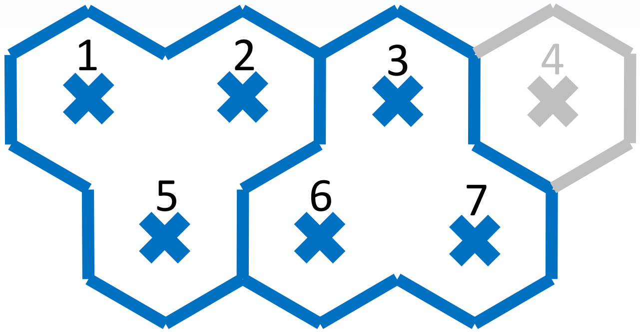
Possible VBSs:

{1}, {2}, {3}, {4},
{5}, {6}, {7},
{1,2,5}, {3,6,7},
{2,5,6}, {3,4,7}

Our Solution: Structuring Set of Possible VBSs

Theorem: If the set of possible VBSs is **2-decomposable**, the maximum weight exact cover problem is solvable in polynomial time.

E.g.



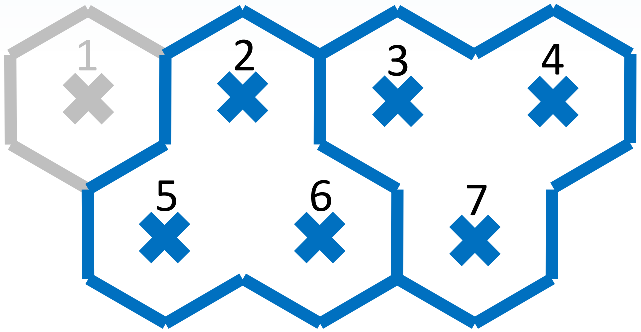
Possible VBSs:

{1}, {2}, {3}, {4},
{5}, {6}, {7},
{1,2,5}, {3,6,7},
{2,5,6}, {3,4,7}

Our Solution: Structuring Set of Possible VBSs

Theorem: If the set of possible VBSs is **2-decomposable**, the maximum weight exact cover problem is solvable in polynomial time.

E.g.

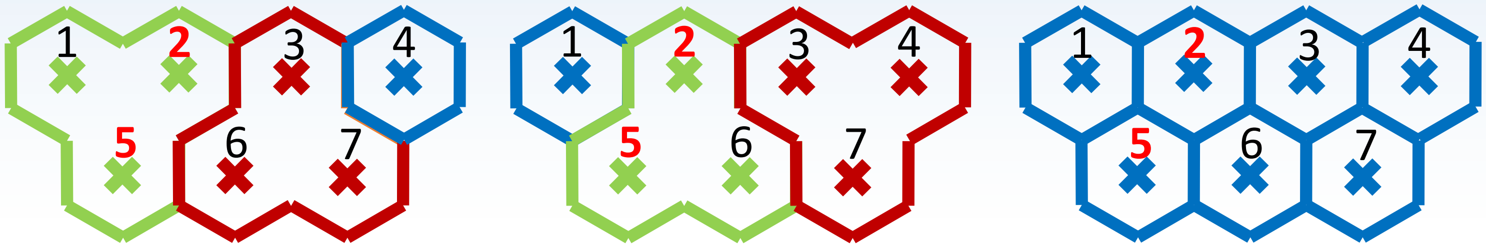


Possible VBSs:

{1}, {2}, {3}, {4},
{5}, {6}, {7},
{1,2,5}, {3,6,7},
{2,5,6}, {3,4,7}

What is 2-Decomposable?

E.g.



...

Set of possible VBSs → Possible VBS partitions:

- Two RRs are **equivalent**, if in any partition,

  OR < They are in the same VBS
  Each of them is a VBS.

- Such relation partitions all RRs into **equivalence classes**

- If each VBS covered by ≤ 2 equivalence classes, the set of possible VBSs is **2-decomposable**.

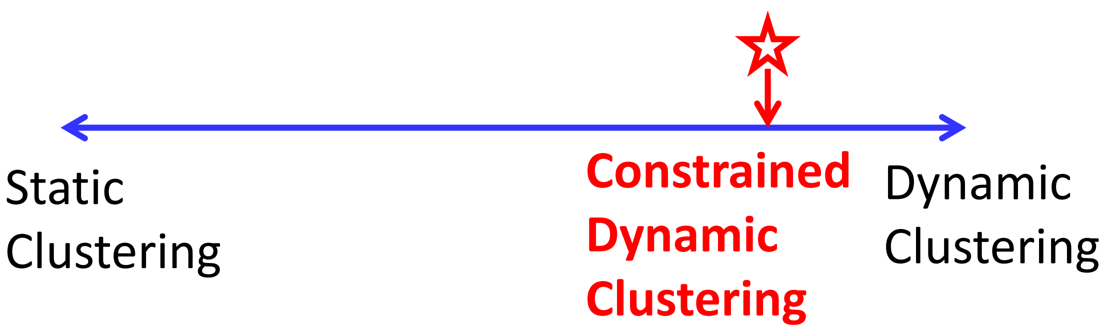
If the Set of VBSs is 2-Decomposable

Theorem: If the set of possible VBSs is **2-decomposable**, the maximum weight exact cover problem is solvable in polynomial time.

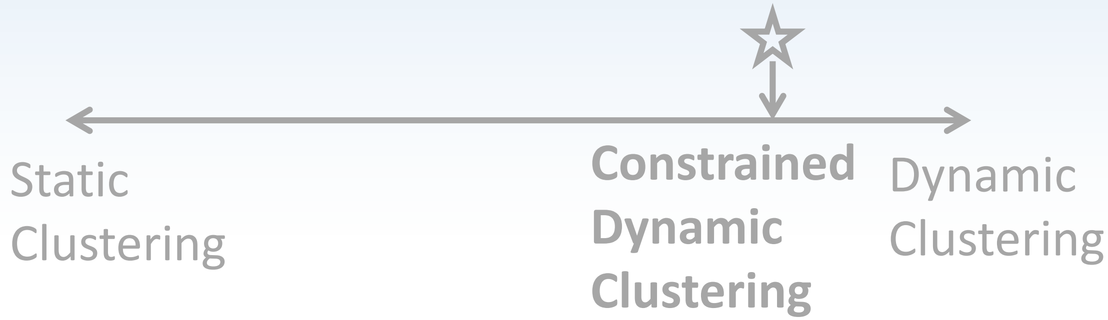
Maximum Weight Exact Cover

⇕

Maximum Weight Matching in a general graph



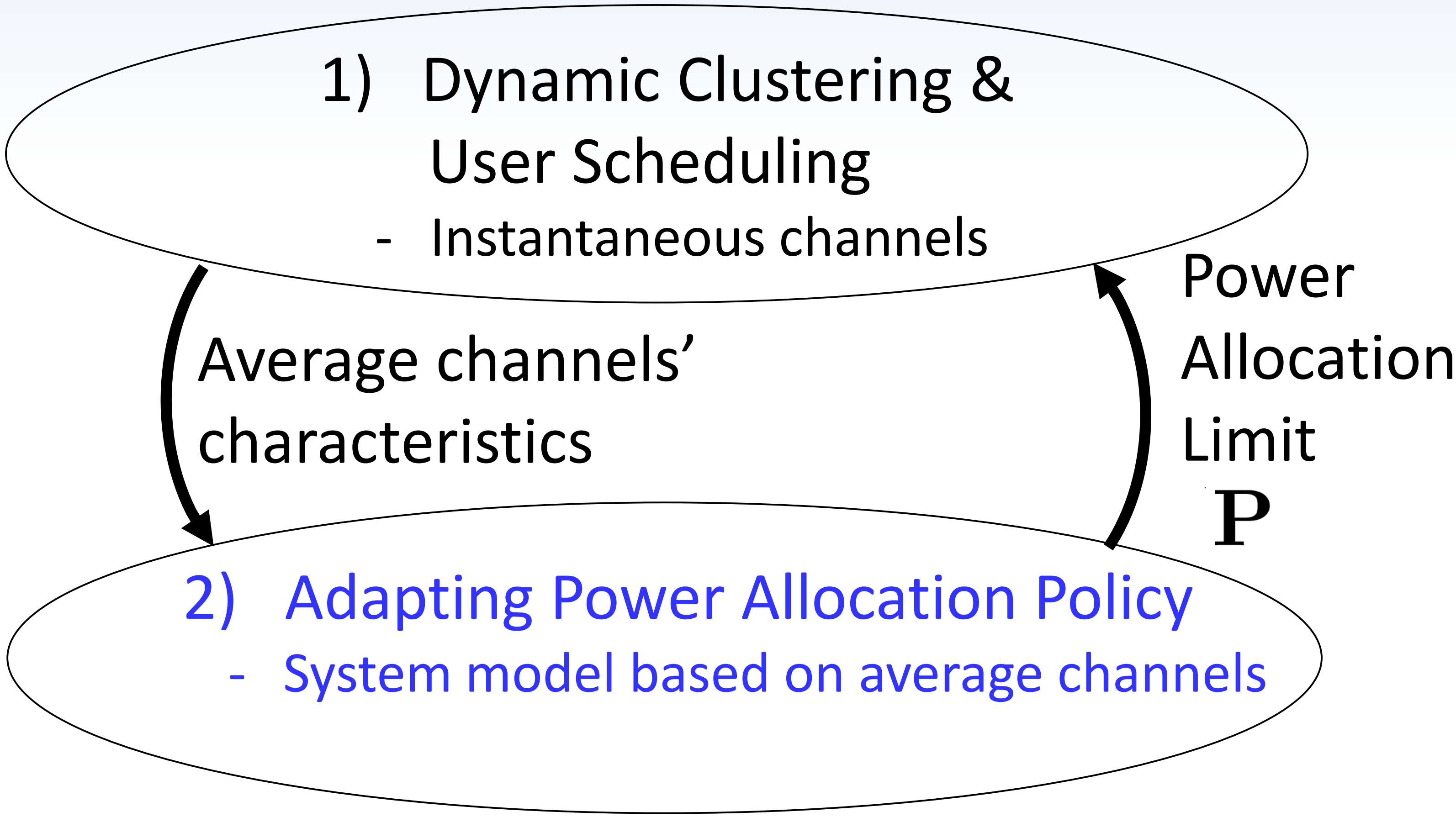
How Limiting are 2-Decomposable Structures for Cellular Systems?



In cellular context

- VBSs of interest are of limited size.
- VBSs of interest include RRs that are close by.
- We can design a 2-decomposable set of VBSs where all RRs can work alone .

Framework



1) Dynamic Clustering & User Scheduling
- Instantaneous channels
Average channels' characteristics
Power Allocation Limit
$\mathbf{P}$
2) Adapting Power Allocation Policy
- System model based on average channels

Adapting Power Allocation Limit on a Slower Time Scale

Inputs:
- Utility fn
- Avg channels
- $\mathbf{P}$/ RR & sub-band

Ouputs:
- New $\mathbf{P}$/ RR & sub-band

depends on avg rate

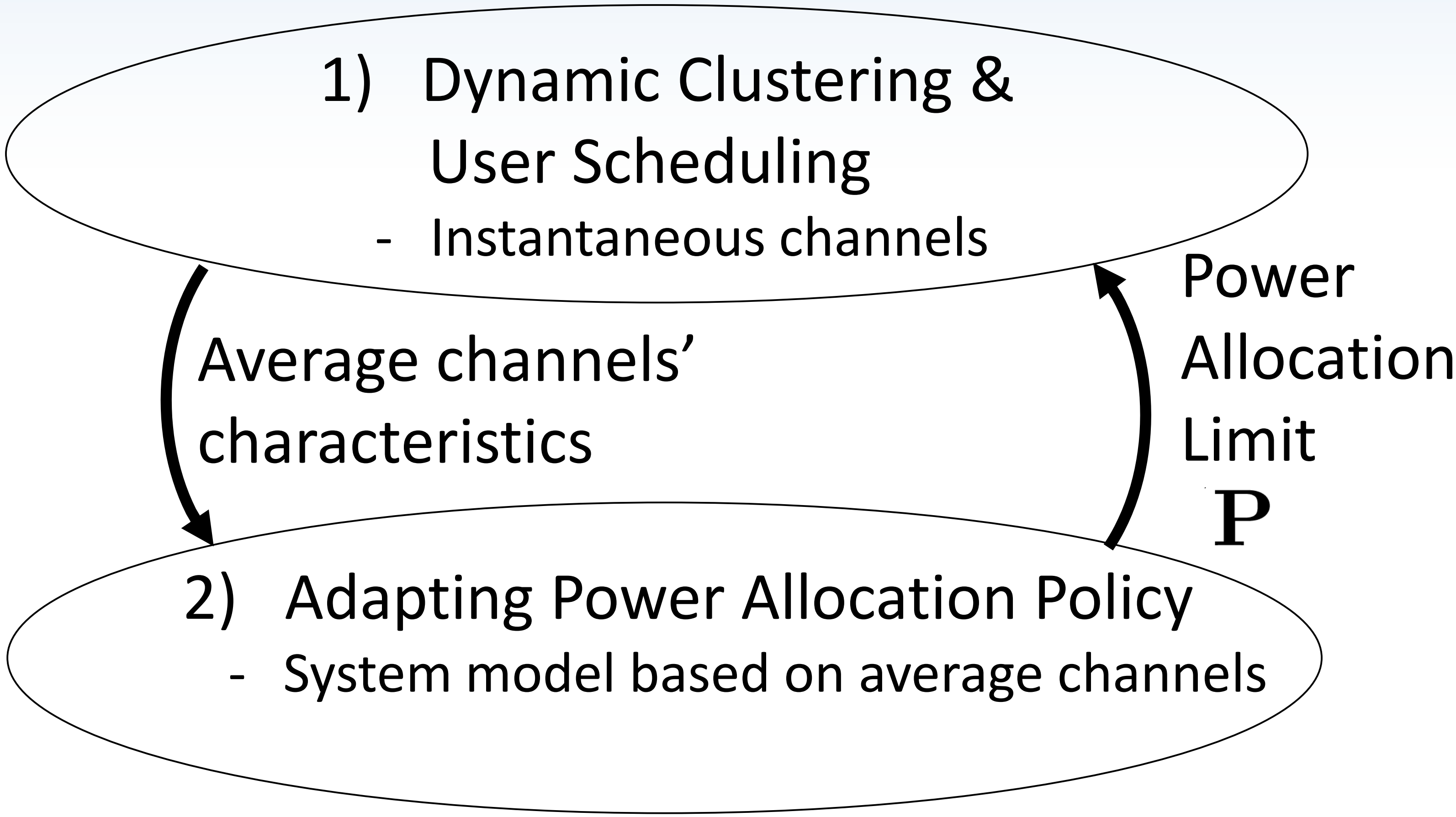
We want to compute sensitivity of max system utility to $\mathbf{P}$

We introduce a virtual system* which is based on average channels.

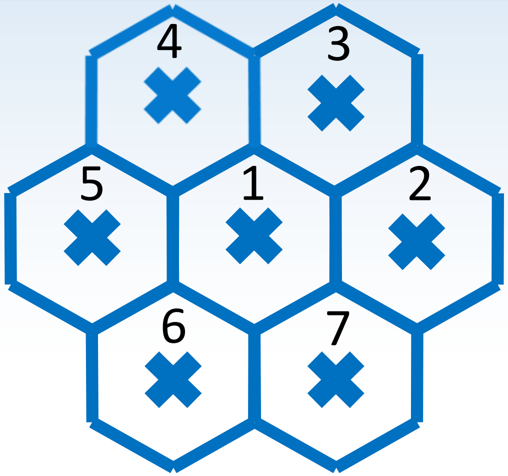
- Captures the time fraction of each decision.

*Based on ideas used in prior work by Stolyar et al.

Framework


1) Dynamic Clustering & User Scheduling
- Instantaneous channels
Average channels' characteristics
Power Allocation Limit
P
2) Adapting Power Allocation Policy
- System model based on average channels

Simulation Setup



- 7 RRs (2 antennas / RR)
- 100 Users (not moving)
- Flat and fast fading
- CoMP used
- A 2-decomposable set of VBSs
- Objective:

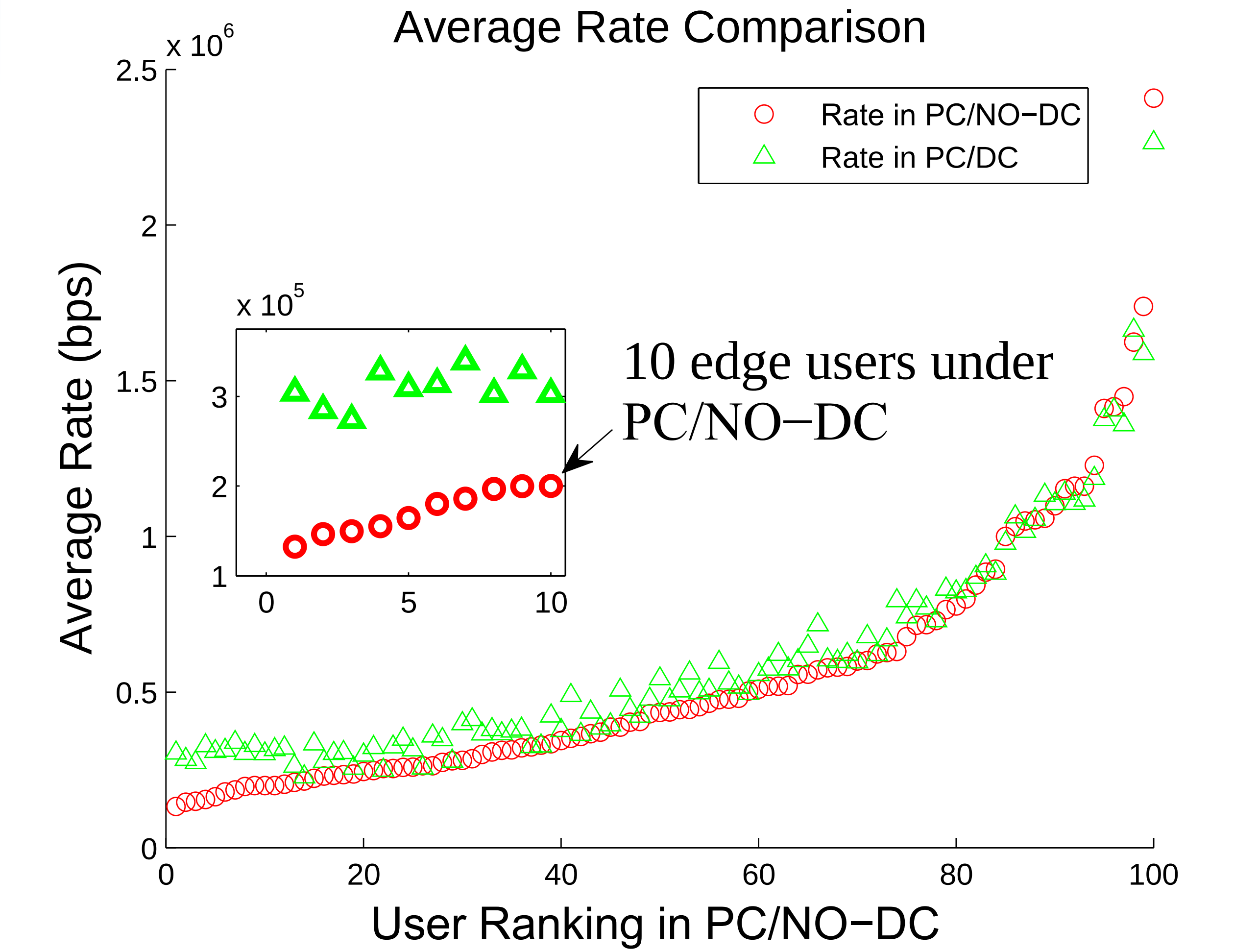
$$\max \sum_i \log(\boxed{X_i})$$

average rate for user i

- Baseline:
  - No Dynamic Clustering (NO-DC)
  - Power Control (PC): adapting power across RR & subband
  - Multi User MIMO scheduling
  - **A very aggressive baseline!**

Traditional Edge User Improvement

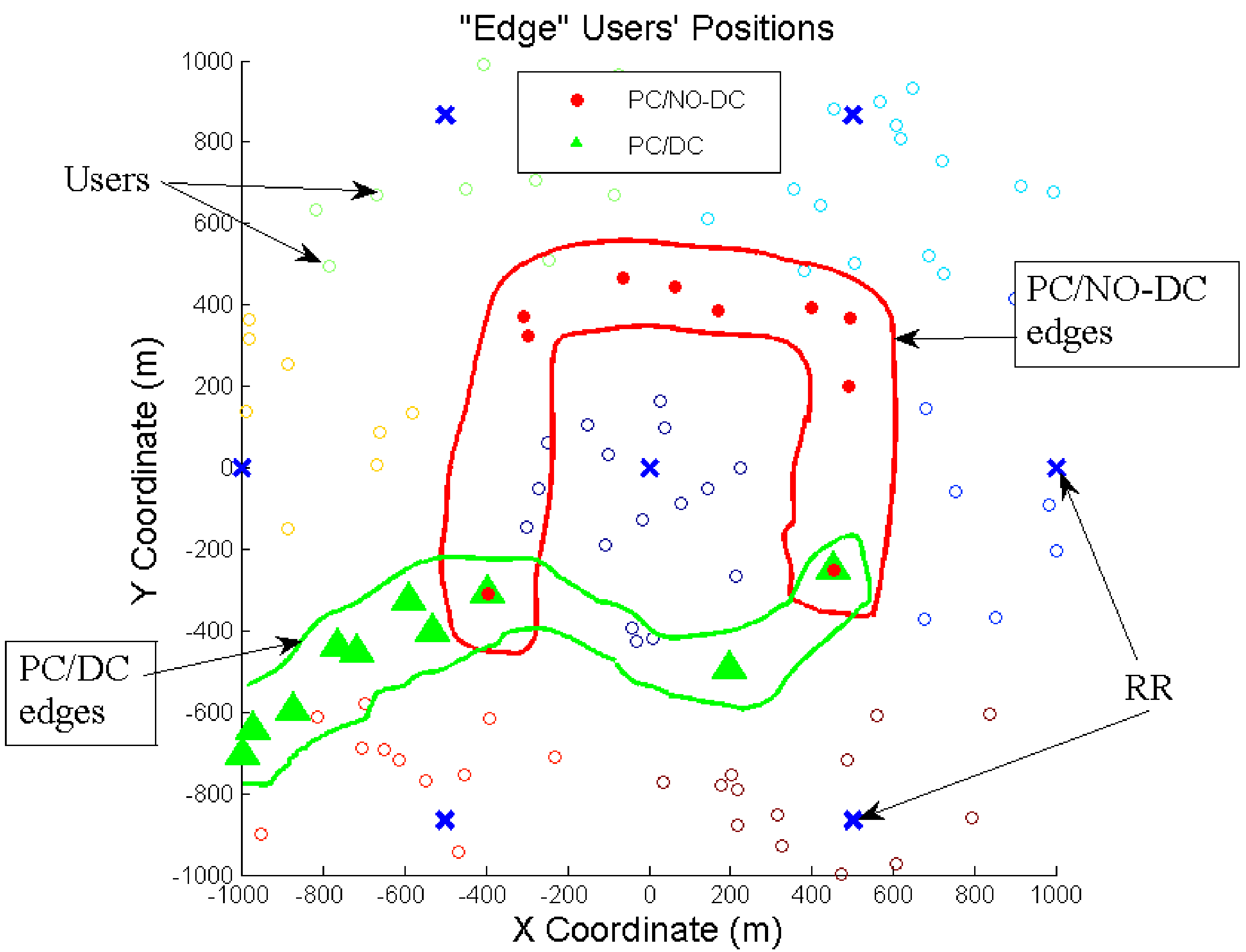
Traditional edge user: worst 10% of users under PC/NO-DC



Throughput Increase of 80% for edge users

Where is edge now?

Moving Edges



Let worst 10% users' positions represent edges

Main Take Away

- Cloud-RAN provides same benefits of other cloud-based systems, and potentially enables cooperation.

- We deal with complexity of joint clustering and user scheduling BY structuring the solution space
  - 2-decomposable

- Performance benefits for “edge users” can be substantial

Backup Slides

Computing Weight (Marginal Utility) for Each VBS

Weight ← scheduled users and power per user

- Exponential # choices for users and power

Our solution:

- Based on ZFBF, we propose greedy **orthogonal** user selection
- Optimal power allocation for scheduled users given constraints $\mathbf{P}$
- Works well in dense networks

Optional: What If We Have More Antennas?

Num of users scaled with num of antennas/RR

	Edge Users' Throughput Gain	All Users' Throughput Gain
2 ant/RR, 100 Users	80%	10%
4 ant/RR, 200 Users	146%	17%
8 ant/RR, 400 users	61%	3%

Comments:

- Recall we are comparing to a very aggressive baseline
- We conjecture that with large # of antennas & users we need to be more careful in scheduling too many users.