A Fast Power Network Optimization Algorithm for Improving Dynamic IR-drop

Speaker: Yang-Tai Kung
Dep. of EE, National Cheng Kung University

Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Introduction

- Powerplanning becomes a more critical step.
  - A modern chip consumes more power when its design complexity continually increases.
  - A design has a small margin for voltage drop as technology process advances.

- Most of previous works only consider static power consumption without handling dynamic power consumption during powerplanning or power network optimization.

- For simplicity, existing studies either adopt the worst or average power consumption when dynamic IR-drop is considered.
  - However, it may waste unnecessary routing resource if they adopt the **worst power consumption** to perform voltage analysis.
  - It still has chance to violate the IR-drop or EM constraints if they adopt the **average power consumption.**

Introduction

- Power network optimization considering dynamic IR-drop is very difficult and is very time consuming without a good strategy.
  - Power consumption of a device may be various at different time and it is composed of a large number of Power consumption Files (PFs for short) during a period of time.
  - It is impossible to optimize power network according to allPFs.

- There exist limited researches about power network optimization for dynamic IR-drop.

- An efficient and effective power optimization approach to consider dynamic IR-drop is required by industry.

Our Contributions

- Optimize power network according to critical PFs selected from a large set of PFs.
  - Apply K-clustering algorithm to classify all PFs into several groups according to power distribution maps (PDMs for short) induced by different PFs .
  - Generate a representative power consumption file (RPF for short) for each group to represent the PFs in the group.

- Propose an efficient and effective approach to repair voltage violations in the hotspot region of power network which are induced by many PFs.

- The experimental results have shown that the ratio of IR-drop violations can be significantly reduced by our methodology compared to the classic approach.

Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

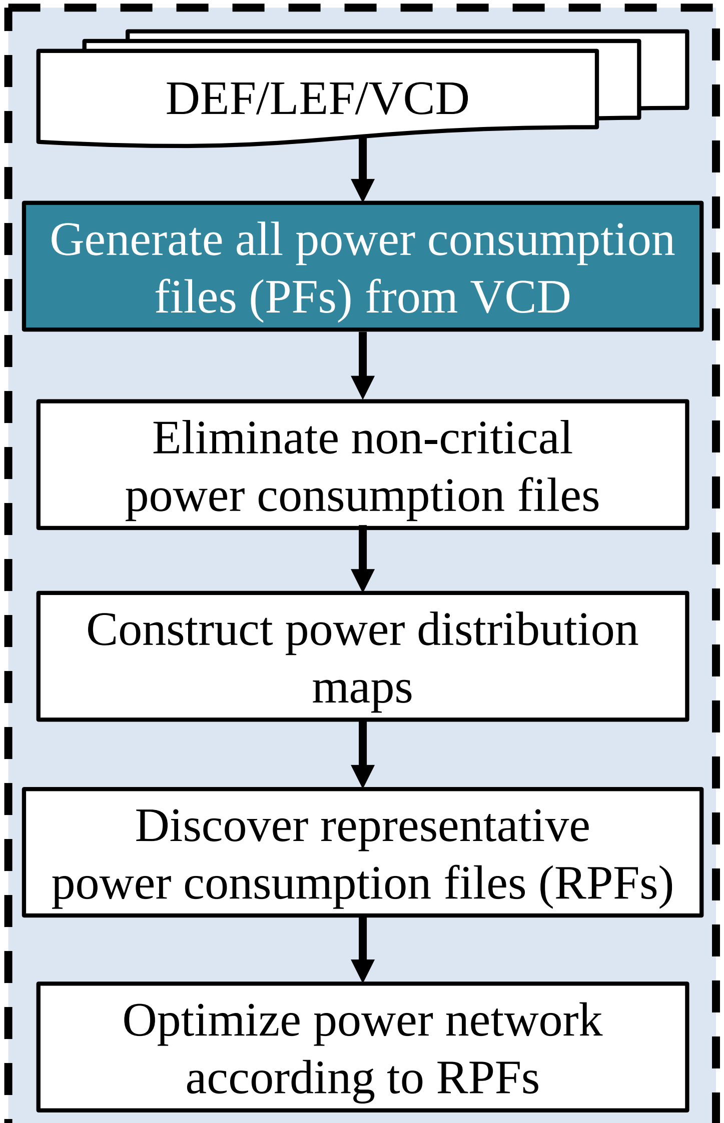
Problem Formulation

- Input:
  - Locations and shapes of standard cells and macros from DEF and LEF files.
  - The average power consumption of standard cells and macros from PTPX (primetime).
  - The capacitances of pins and wires and the power network of the chip extracted from IC Compiler.
  - Switching activity information from RTL Value Change Dump (VCD for short).
- Output:
  - The widths of vertical power stripes (VPSs for short).
- Constraints:
  - The IR-drop constraint
  - The maximum current density (EM) constraint
  - The minimum width constraint
  - The maximum width constraint

Outline

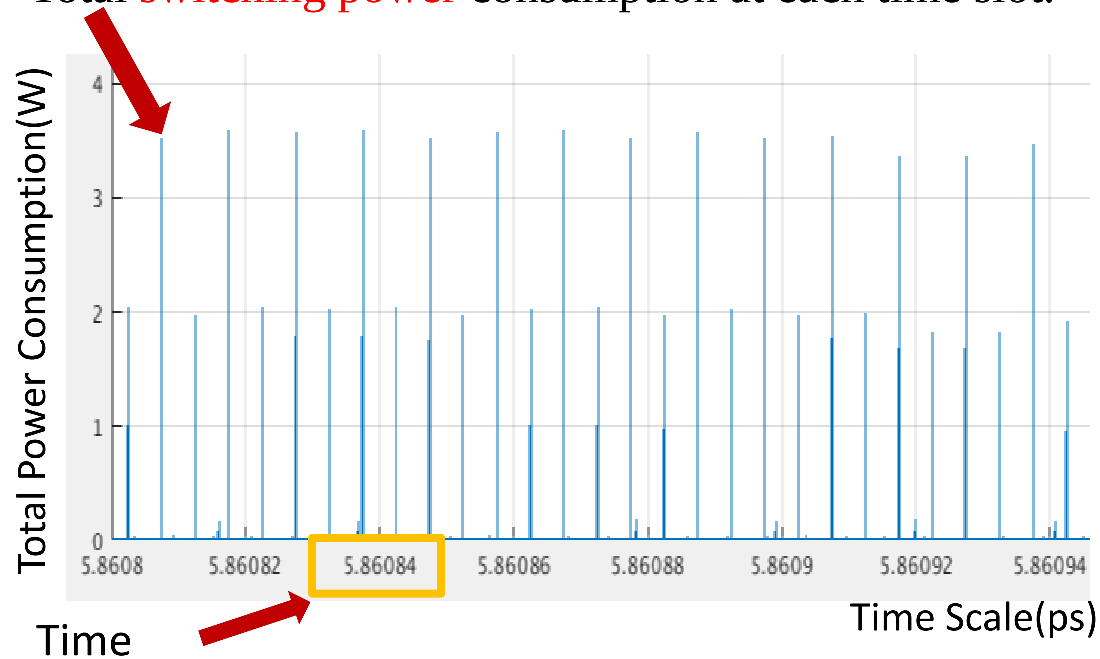
- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Overview of Our Methodology

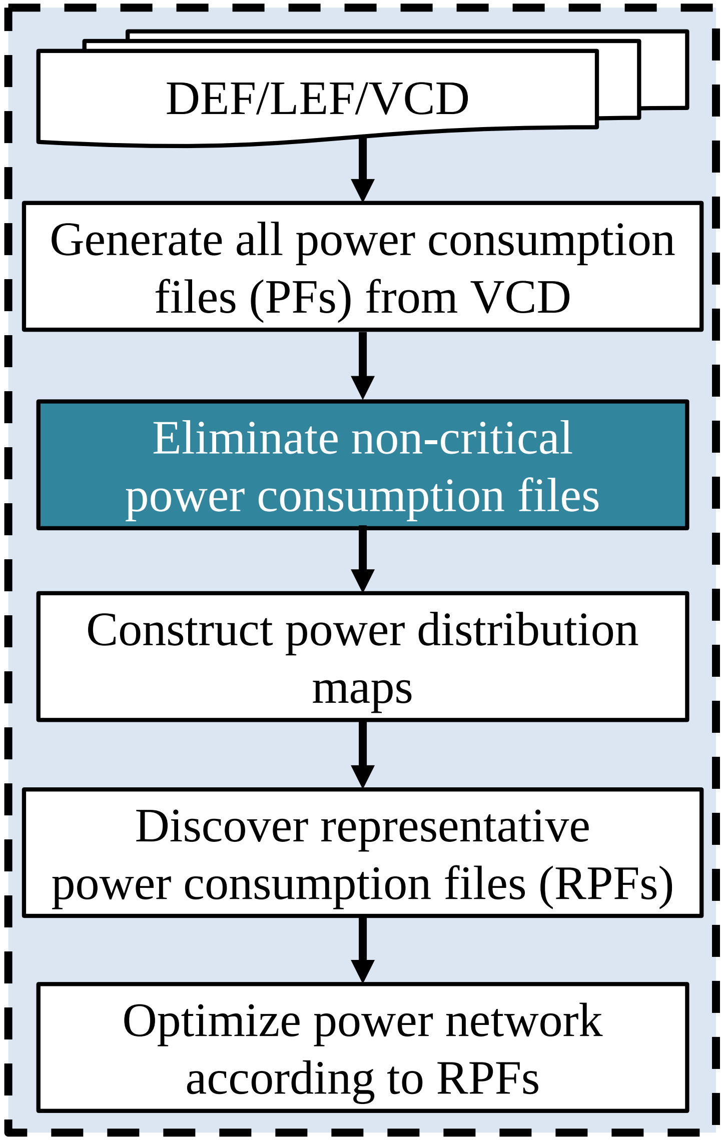


- Generate PFs according to PrimeTime and VCD file, where the switching power of cells are estimated from signal switching information in a VCD file for all time slots.
  - $P_{switching} = \frac{1}{2} * C * V^2$

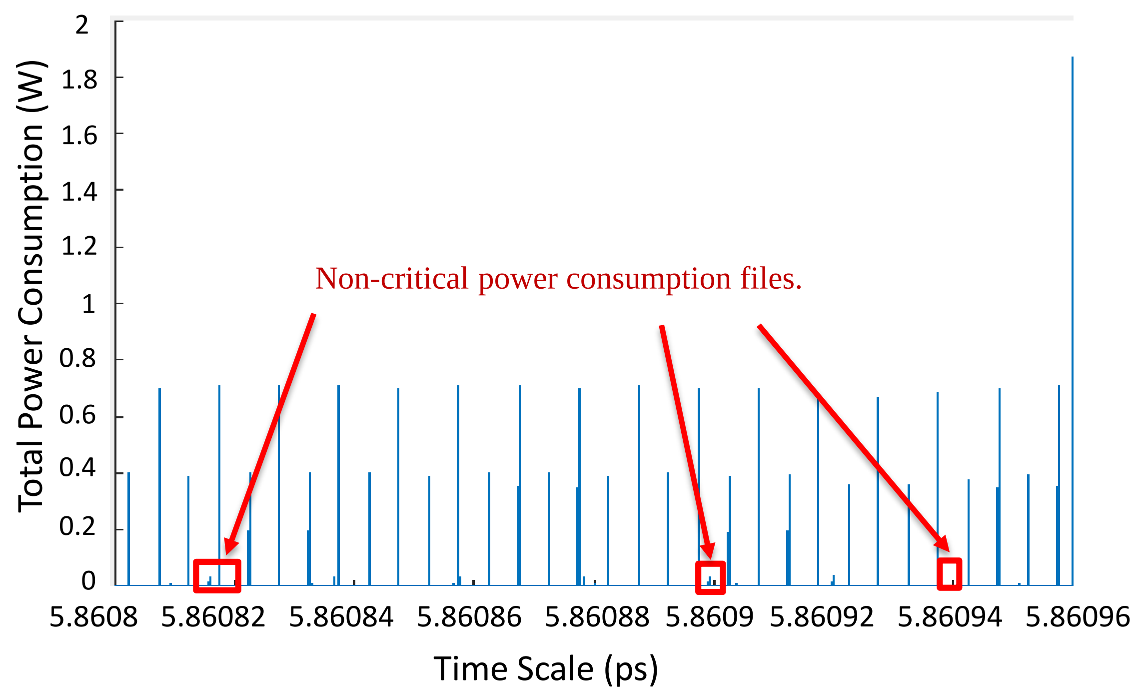
Total switching power consumption at each time slot.

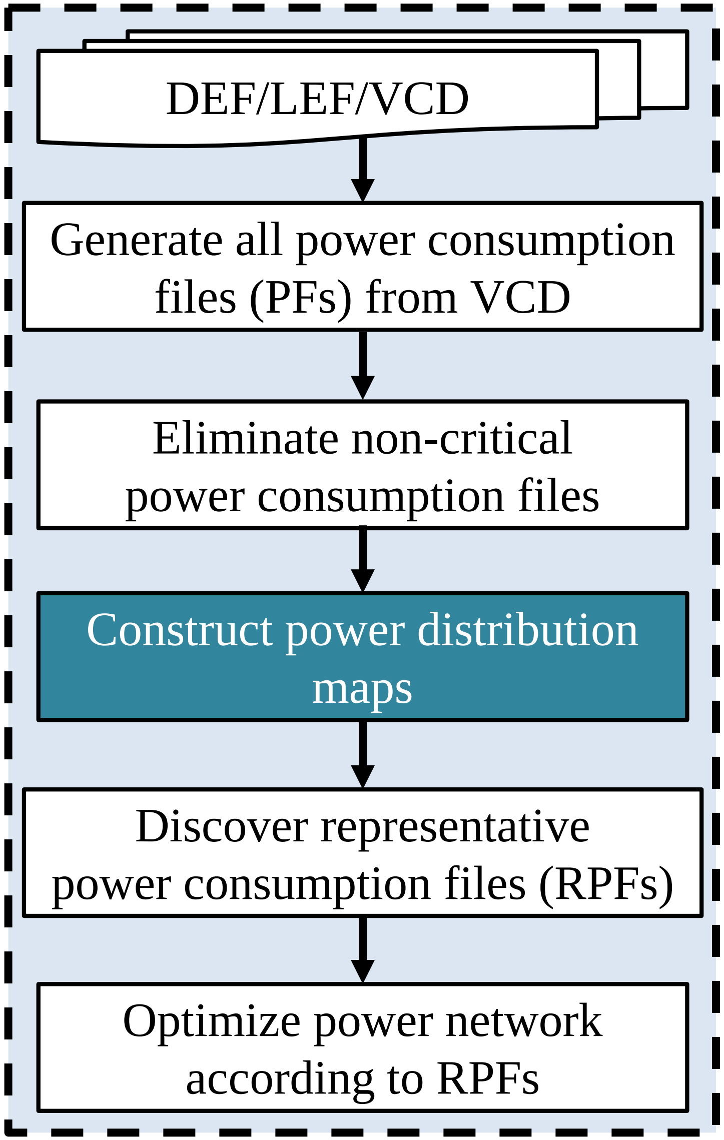


Overview of Our Methodology

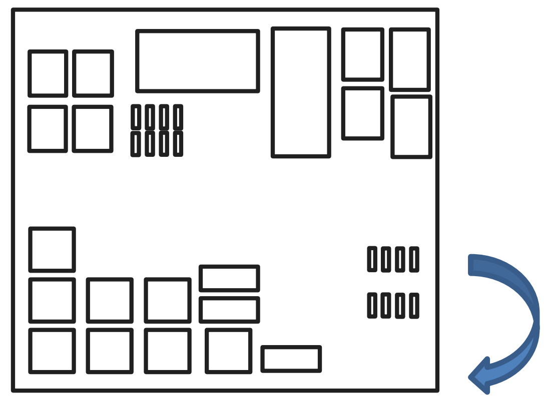


- Eliminate non-critical PFs.
  - A PF is not critical when its total power consumption is smaller than average power consumption.





- Build PDMs for the critical PFs.
  - A map is composed of h grids.
  - A PDM for a PF f_i is represented by a vector X_i= <x_1^i, x_2^i, … x_h^i>, where x_j^i denotes power consumption in a grid j .



Each PF

0.003	0.013	0.017	0.12	0.031
0.006	0.044	0.027	0.021	0.042
0.06	0.065	0.093	0.034	0.095
0.069	0.007	0.002	0.018	0.003
0.01	0.051	0.041	0.033	0.051

PF f_1

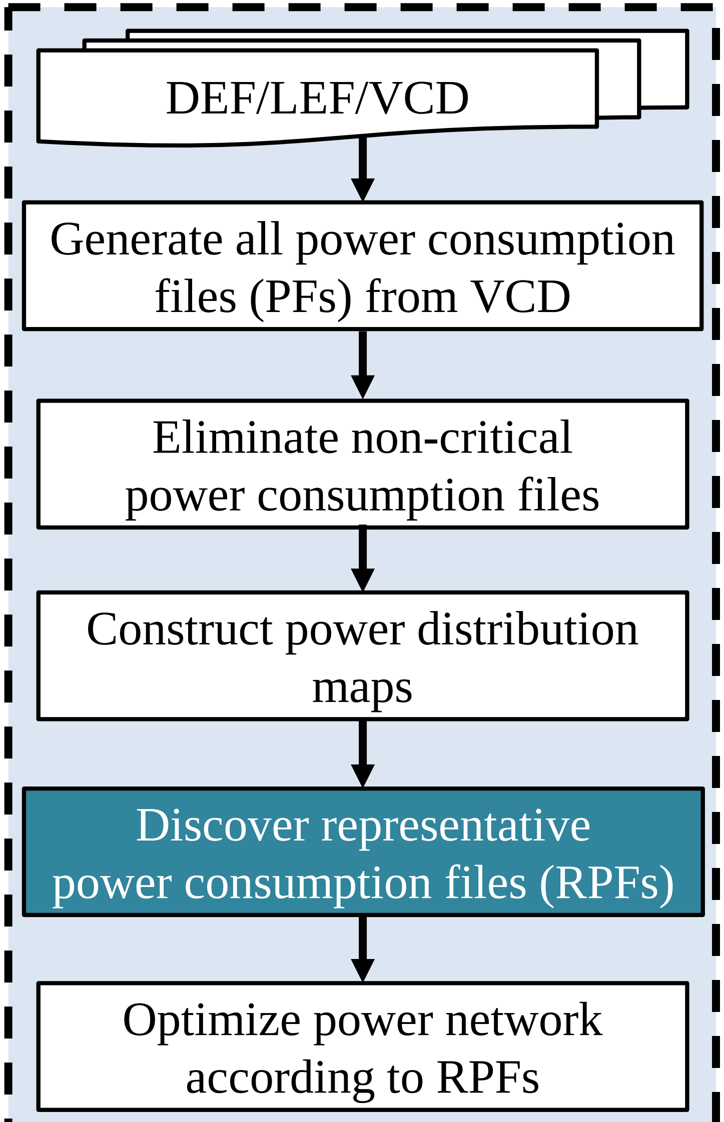
0.03	0.023	0.017	0.12	0.038
0.004	0.034	0.037	0.021	0.012
0.03	0.015	0.071	0.034	0.005
0.022	0.027	0.02	0.019	0.034
0.1	0.051	0.041	0.033	0.021

PF f_2

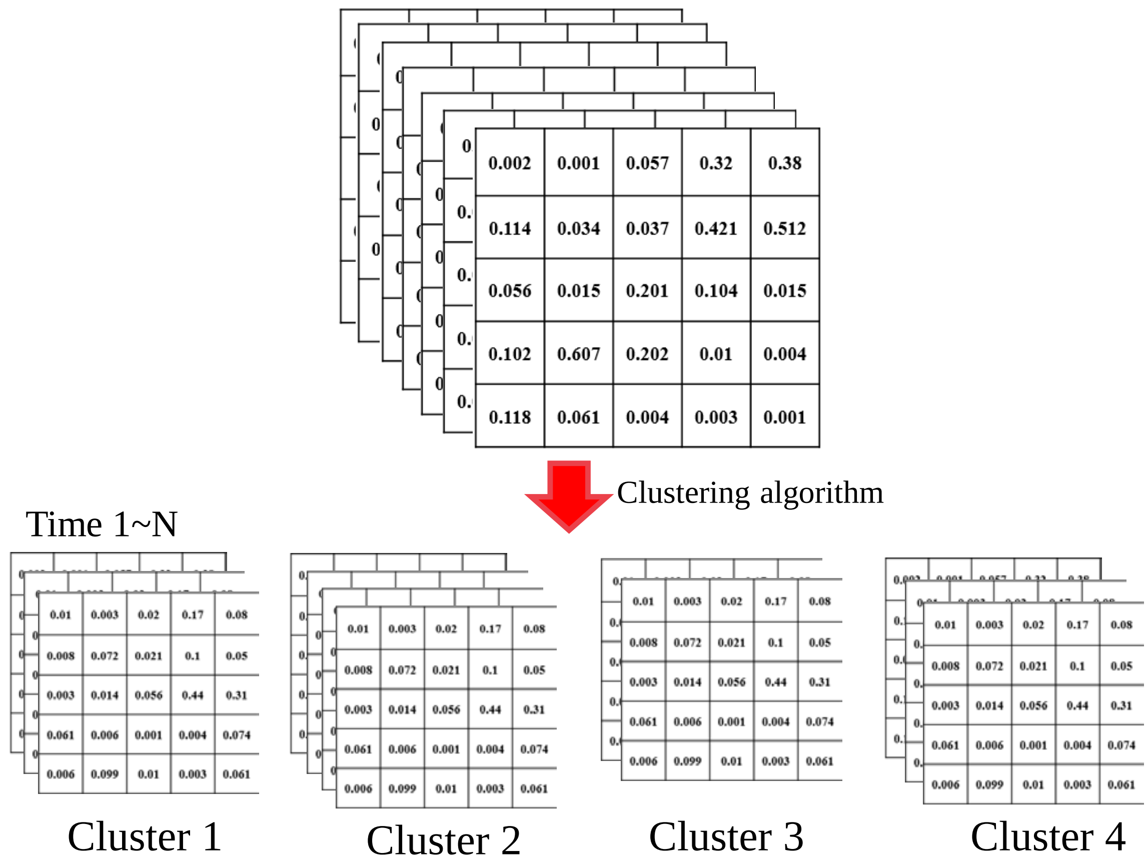
0.002	0.001	0.057	0.32	0.38
0.114	0.034	0.037	0.421	0.512
0.056	0.015	0.201	0.104	0.015
0.102	0.607	0.202	0.01	0.004
0.118	0.061	0.004	0.003	0.001

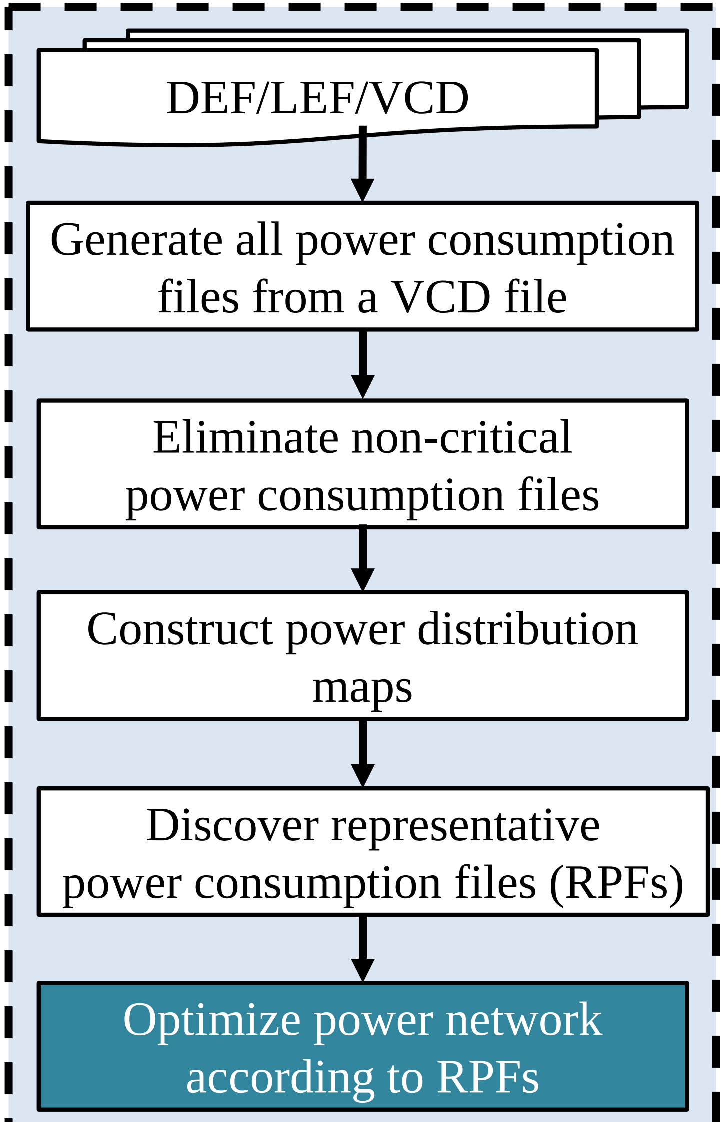
PF f_3

Overview of Our Methodology



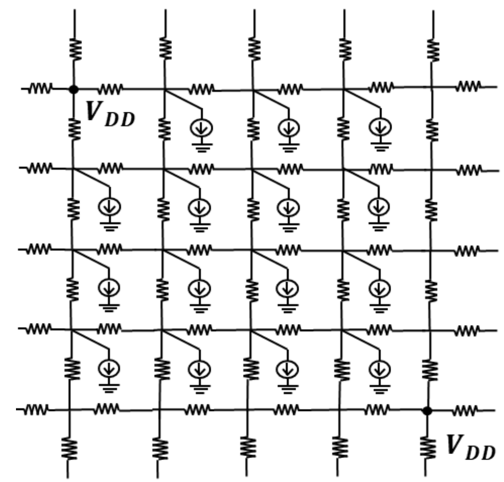
- Classify PFs into different clusters according to their power distributions, and then select a RPF for each cluster.





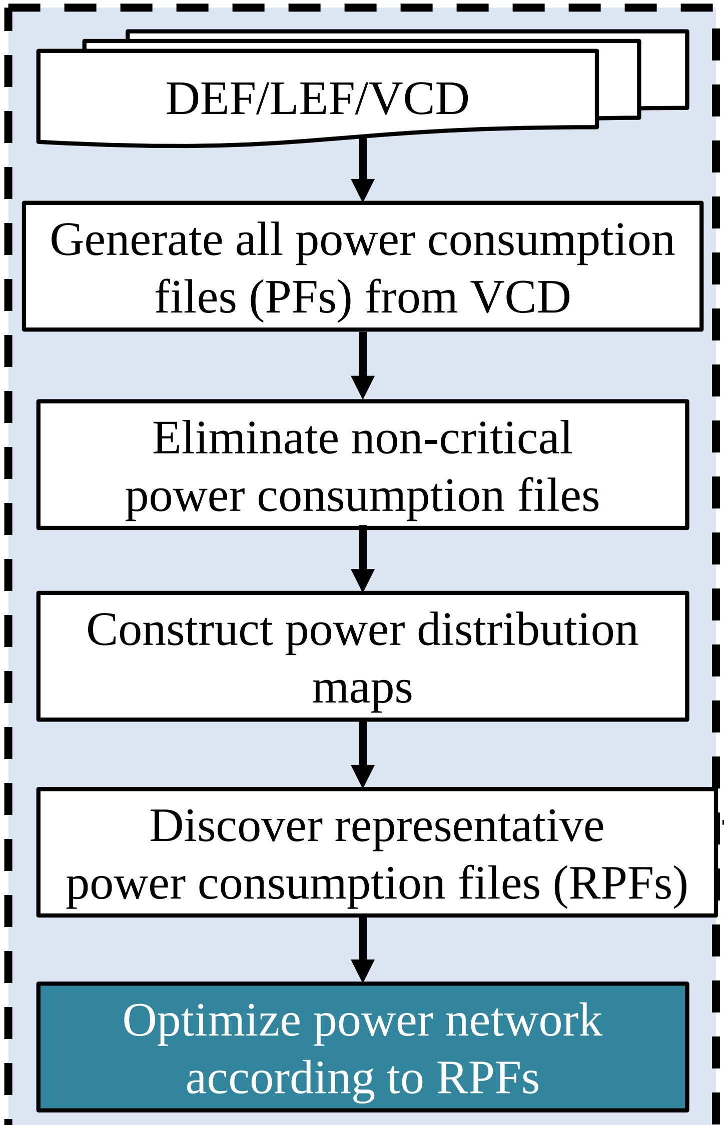
- Construct a voltage violation map (VVM for short) for a PF, where each grid g corresponds to a node in a power network.
  - Positive value in a grid denotes the IR-drop violation value of a node; otherwise, the value is zero.

0.2	**0.3**	**0.1**	**0.1**	**0.1**
0.3	**0.3**	**0.2**	**0.2**	**0.2**
0.3	**0.4**	**0.6**	**0.3**	**0.5**
0.1	**0.1**	**0.3**	**0.3**	**0.4**
0.1	**0.3**	**0.4**	**0.3**	**0.2**

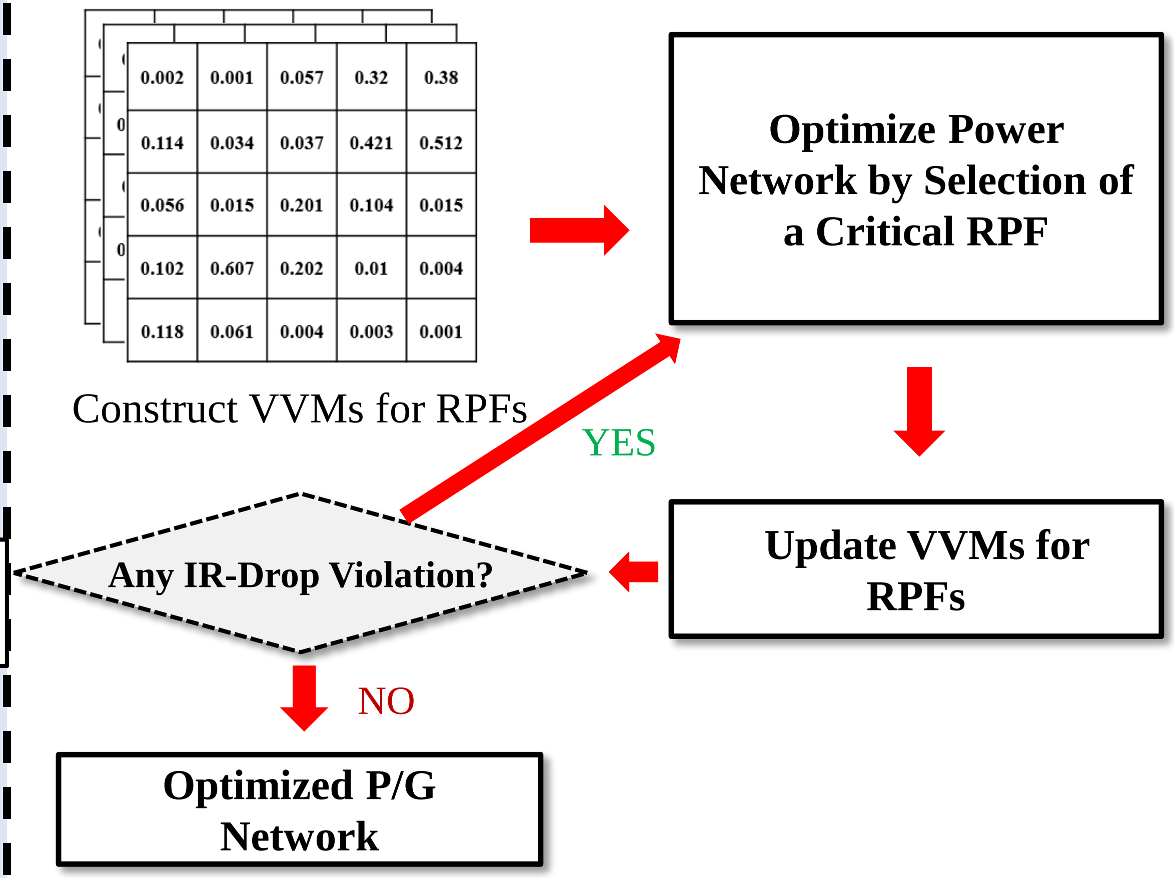


Construct a VVM for a PF

Overview of Our Methodology



- Repeatedly select a critical RPF to optimize power network until the power network has no IR-drop violation for all RPFs.

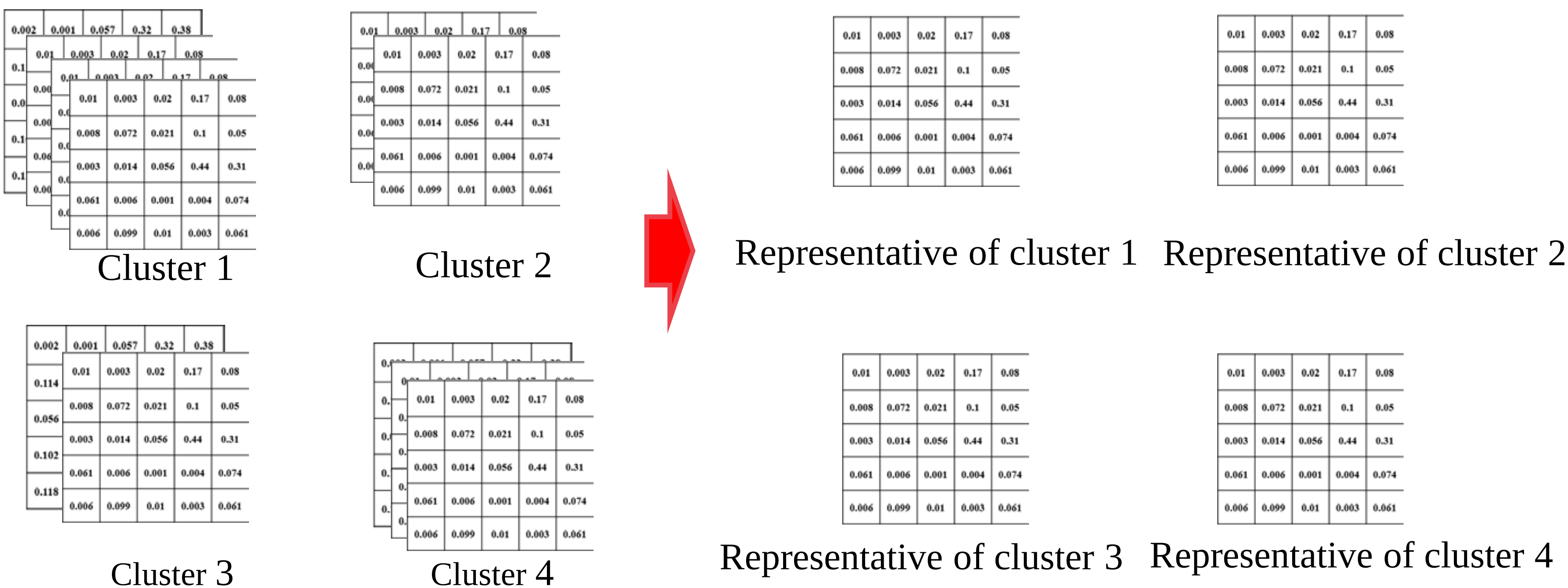


Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
  - Correlation Metric
  - K-Clustering Algorithm
  - Cost Function for a RPF in Each Group
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Discovery of Representative Power Consumption Files

- Discover RPFs and optimize power network according to these files.
  - Runtime can be saved significantly without sacrificing design quality.
- Method:
  - Step1: Cluster PFs according to their power distributions.
  - Step2: Find a RPF for each cluster.



Correlation Metric

- We use distance $d_{i,j}$ to represent the similarity between PDMs X_i and X_j of two PFs f_i and f_j, which is calculated by the following function:

$$d_{i,j} = 1 - \rho_{i,j} \ , \ 0 \leq d_{i,j} \leq 2$$

  - where $\rho_{i,j}$ is called as statistical correlation, which is defined as follows:

$$\rho_{i,j} = \frac{cov(i,j)}{\sigma_i \, . \, \sigma_j} \ , \ -1 \leq \rho \leq 1$$

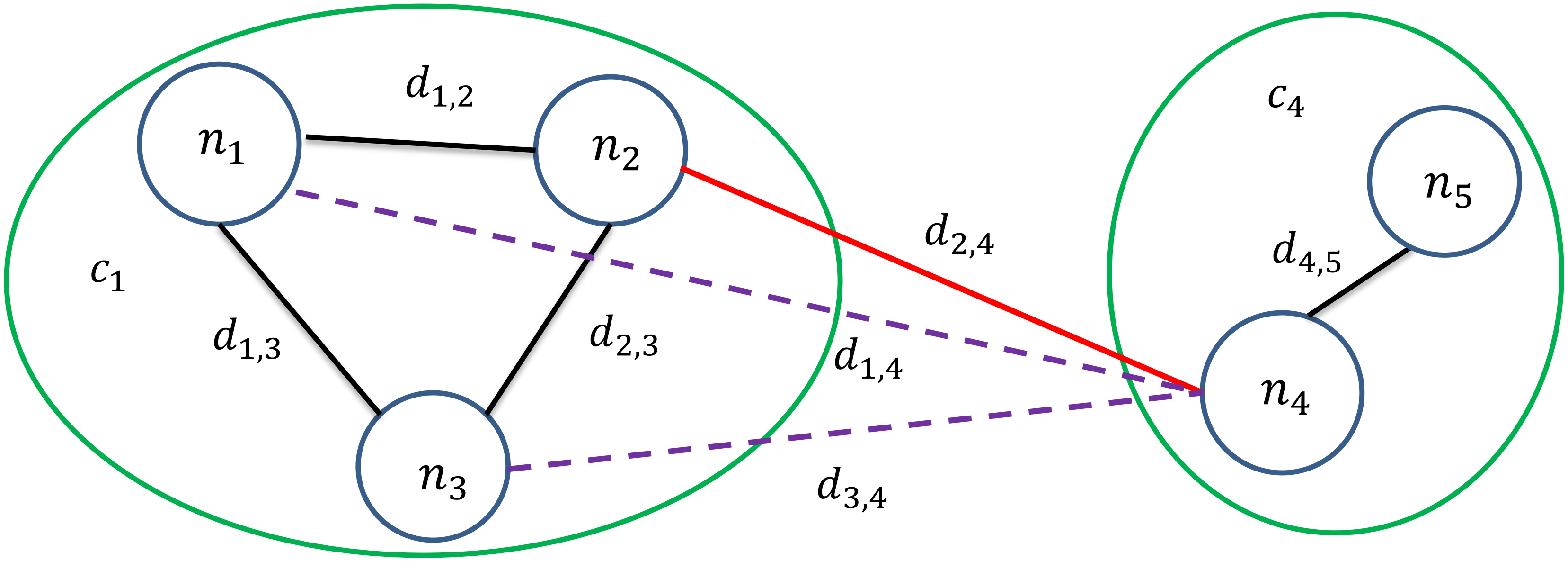
    - where σ_i (σ_j) is standard deviation.
  - function $cov(i,j)$ represents the covariance of PDMs X_i and X_j as follows:

$$cov(i,j) = \frac{1}{h} * \sum_{b=1}^{h} \left[\left(x_b^i - \bar{x}^i \right) * \left(x_b^j - \bar{x}^j \right) \right]$$

    - $\bar{x}^i$($\bar{x}^j$) denotes the average power consumption in the vector X_i(X_j).
    - h denotes the number of bins in a PDM.

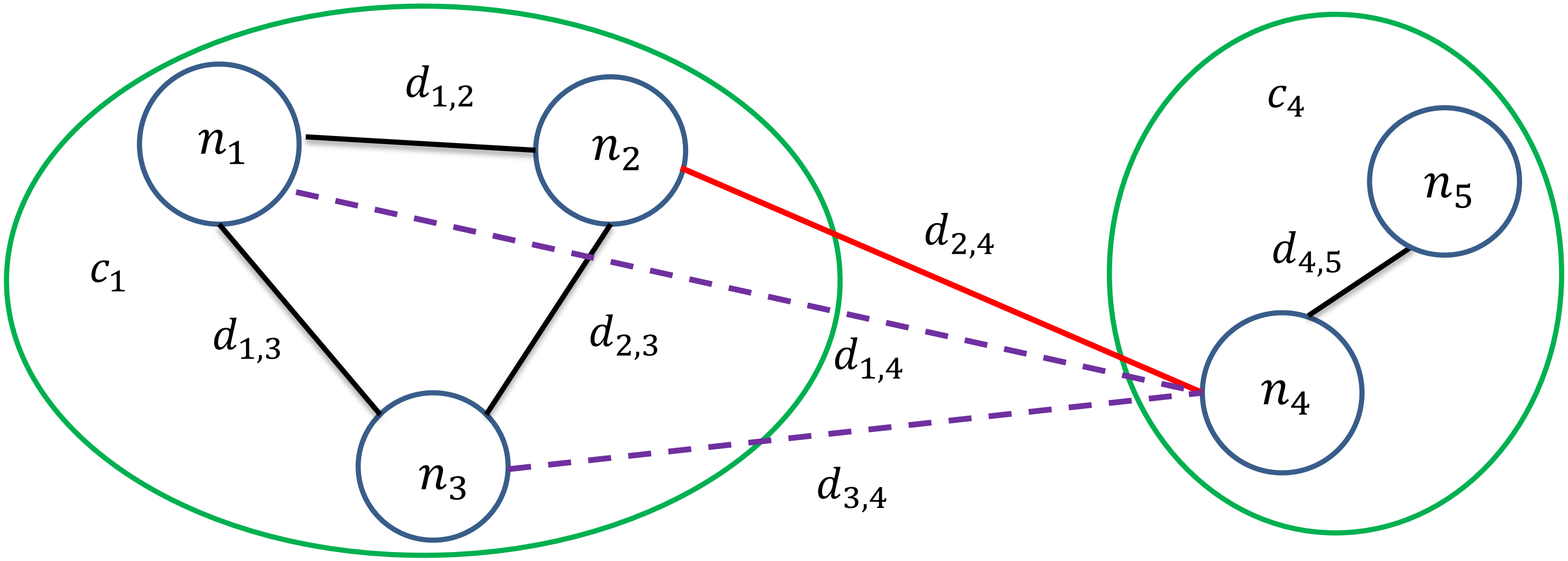
K-Clustering Algorithm

- Divide all PFs into **K** clusters according to their power distributions.
  - PFs in each cluster have similar power distribution.
- Construction of a complete graph *G(N, E)* before applying **K** cluster algorithm.
  - Initialize a node n_i for each PF f_i.
  - Connect an edge $e_{i,j}$ for each pair of n_i and n_j, where $d_{i,j}$ denote the weight of the edge.



K-Clustering Algorithm

- In the beginning, we consider each node n_i as a cluster c_i.
- The algorithm will repeatedly combine two clusters until the number of clusters is **K**.
- After a node n_i is merged into a cluster, we use c_i to represent the associated cluster of n_i for each edge $e_{i,j}$ for simplicity.



K-Clustering Algorithm

Algorithm: K-Clustering Algorithm

Input: a weighted complete graph *G(N, E)*; **K** is number of clusters;
Output: **K** clusters, each cluster has the similar power consumption files;

Sort all edges in *E* in the non-decreasing order and store in *L*.
$S \leftarrow \{\emptyset\}$; ▸ Let *S* denote a set of edges connecting to two clusters.
$\kappa \leftarrow |N|$;
for (each edge $e_{i,j}$ in *L*) **do**
    for (each edge $e_{p,q}$ connecting to c_i *and* c_j) **do**
        $S \leftarrow S \cup \{d_{p,q}\}$;
    end for
    If ($d_{i,j}$ is the smallest value in S) **then**
        $c_i \leftarrow c_i \cup c_j$;
        Remove all edges *S* from *E*;
        $\kappa \leftarrow \kappa - 1$;
    end if
    If (κ = **K**) **then**
        break;
    end If
    $S \leftarrow \{\emptyset\}$;
end for

Discovery of Representative Power Consumption File

- Select a RPF from each cluster $|C|$ if it has the maximum value according to the following function:

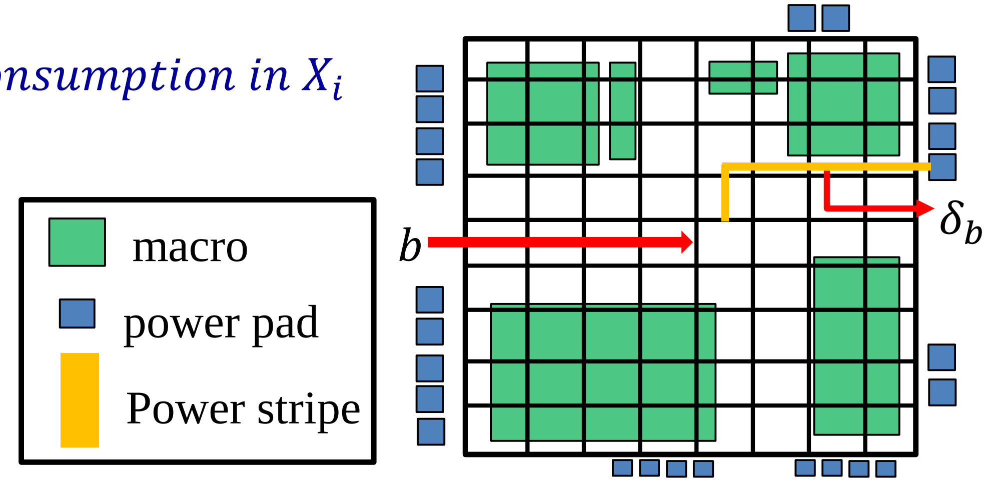
Power consumption

$$\Psi(f_i) = \sum_{b=1}^{h} (\alpha * x_b^i + r_b * \delta_b)$$

Distance weight

- x_b^i : the power consumption of bin b in PF f_i.
- α : the power consumption weight of user defined.
- δ_b: the distance of bin b to the closest power pad.

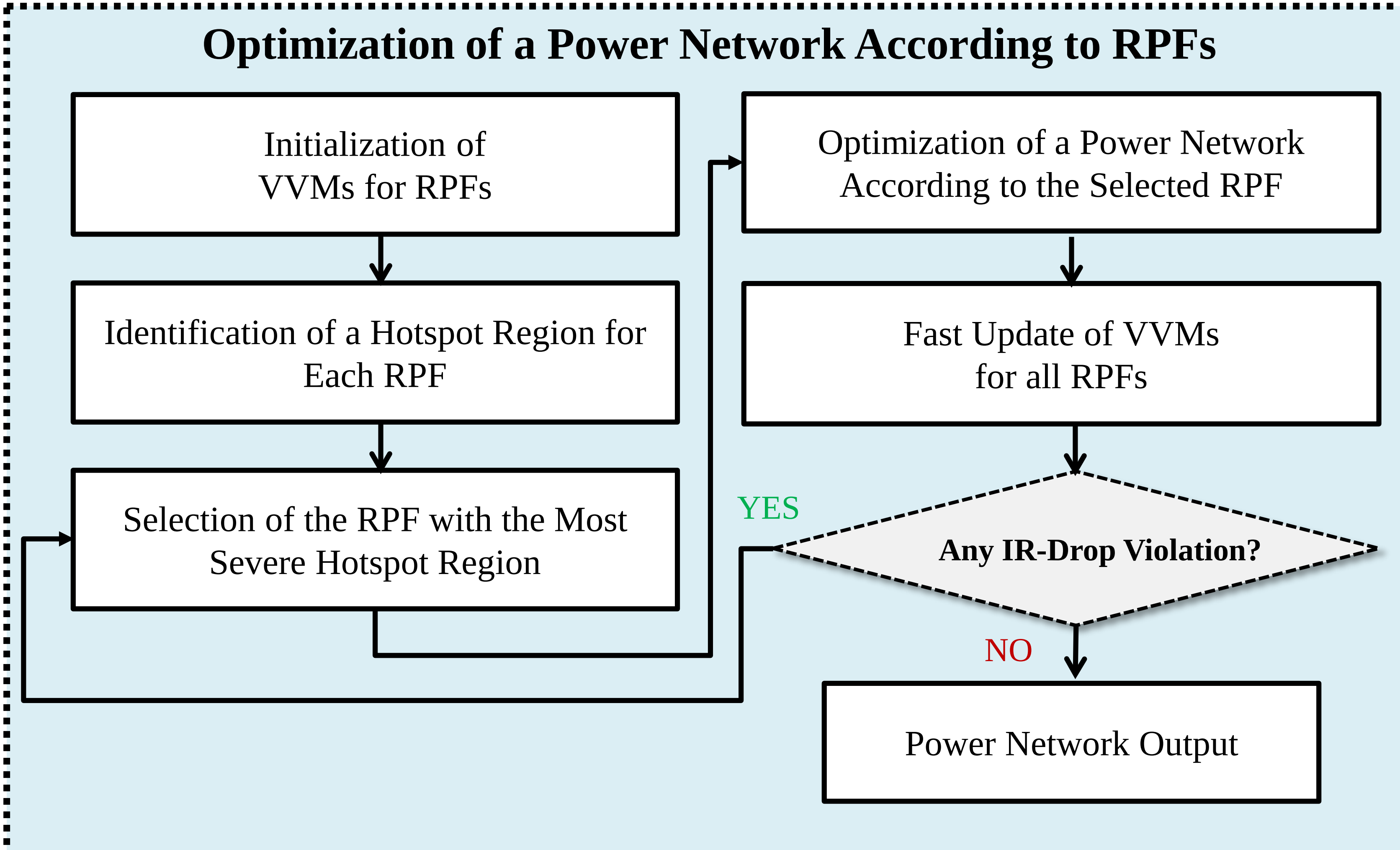
- $r_b = \begin{cases} 1 & , if\ x_b^i\ is\ top\ \eta\%\ power\ consumption\ in\ X_i \\ 0 & , \text{else} \end{cases}$



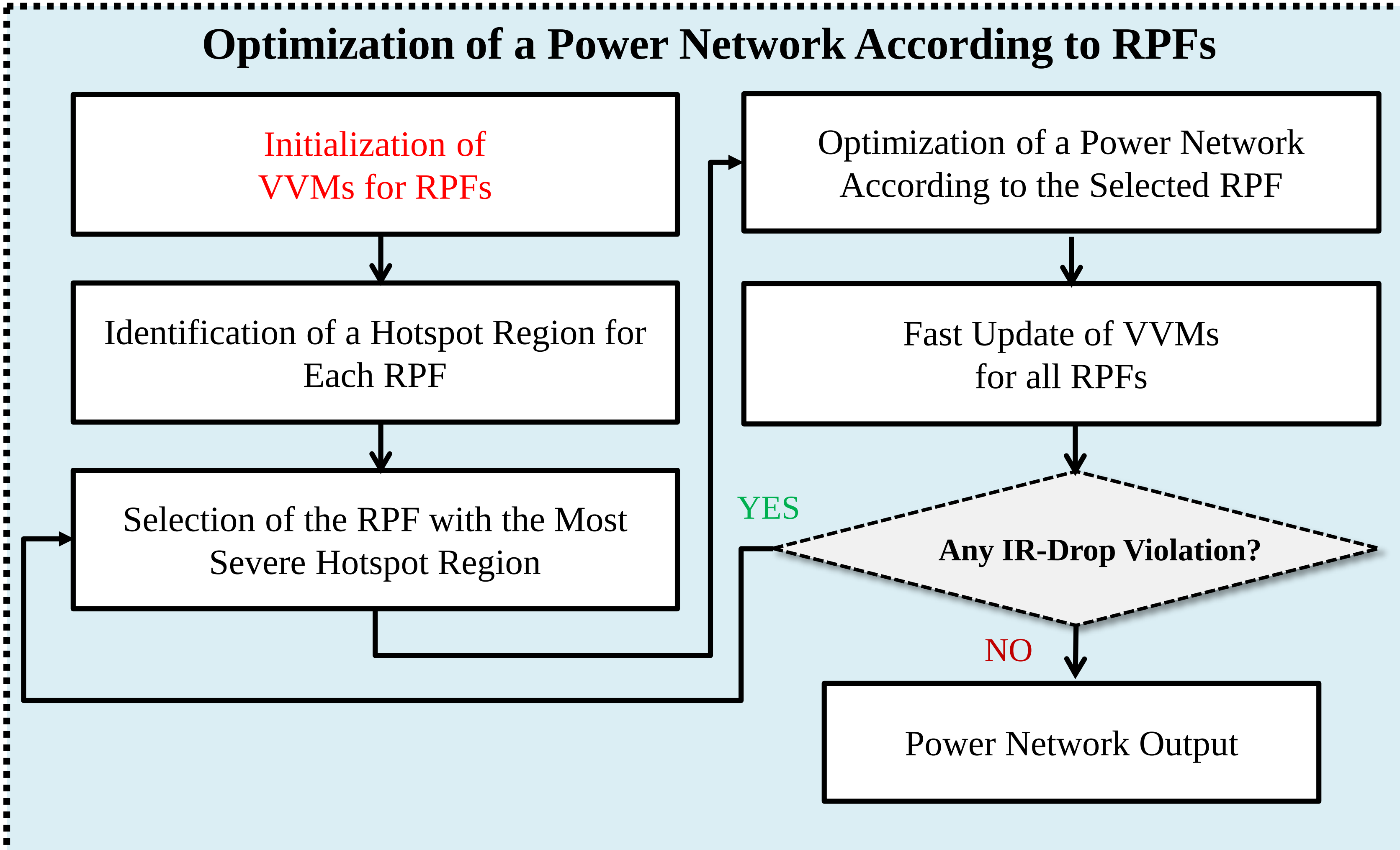
Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

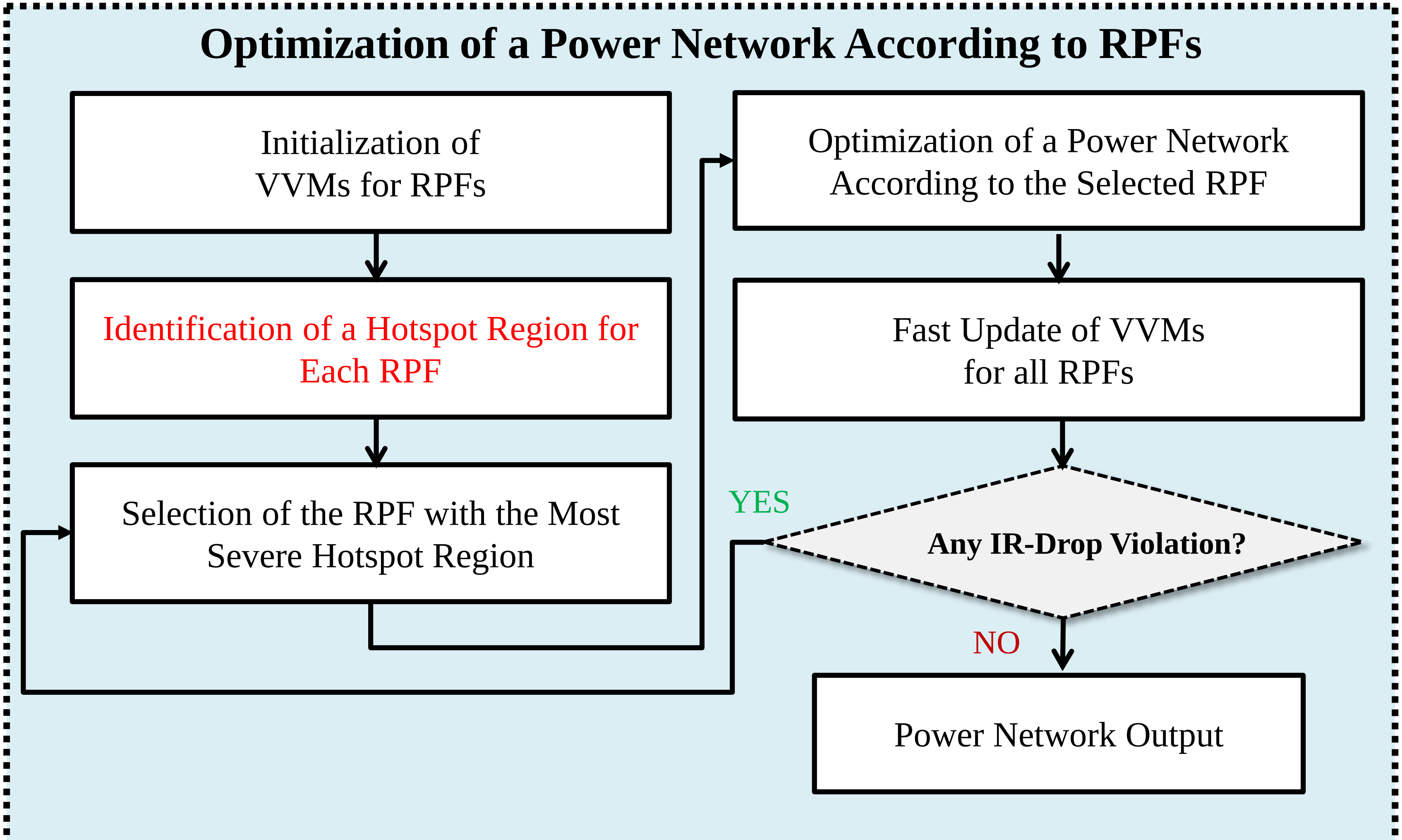
Power Network Optimization Algorithm



Initialization of Voltage Violation Maps for RPFs

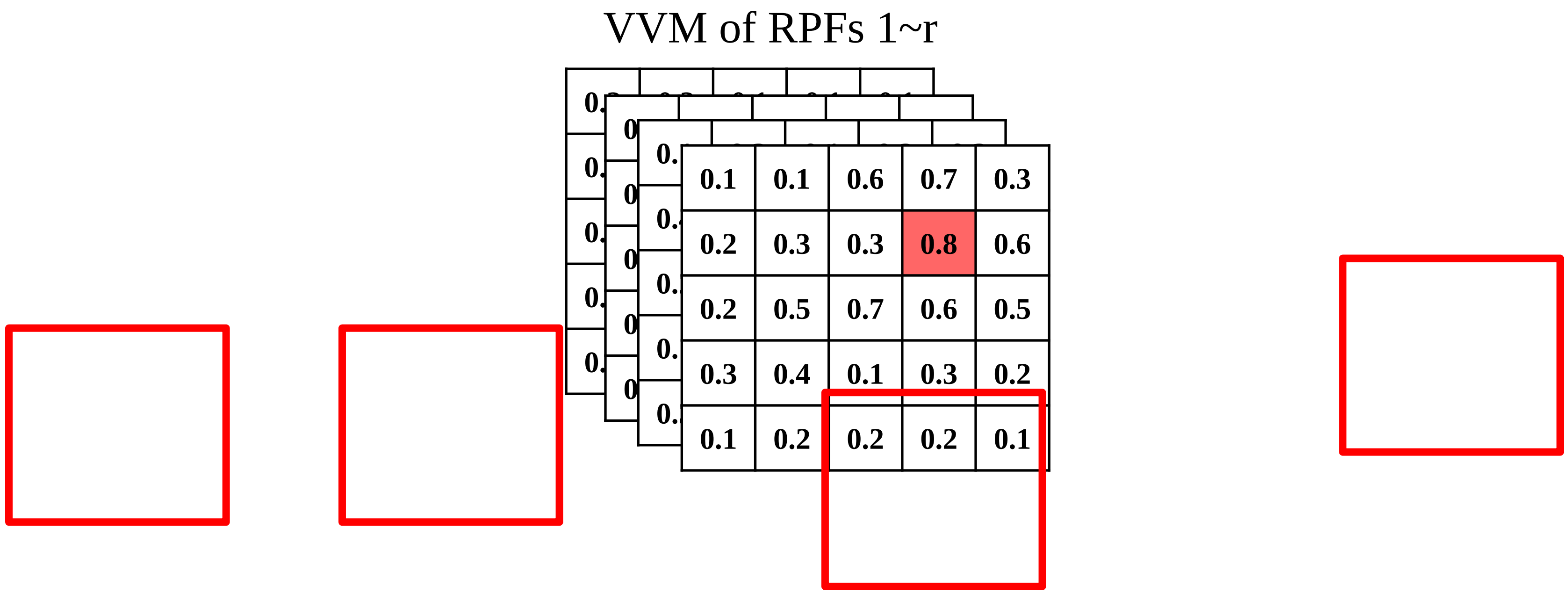


Identification of a Hotspot Region for Each RPP



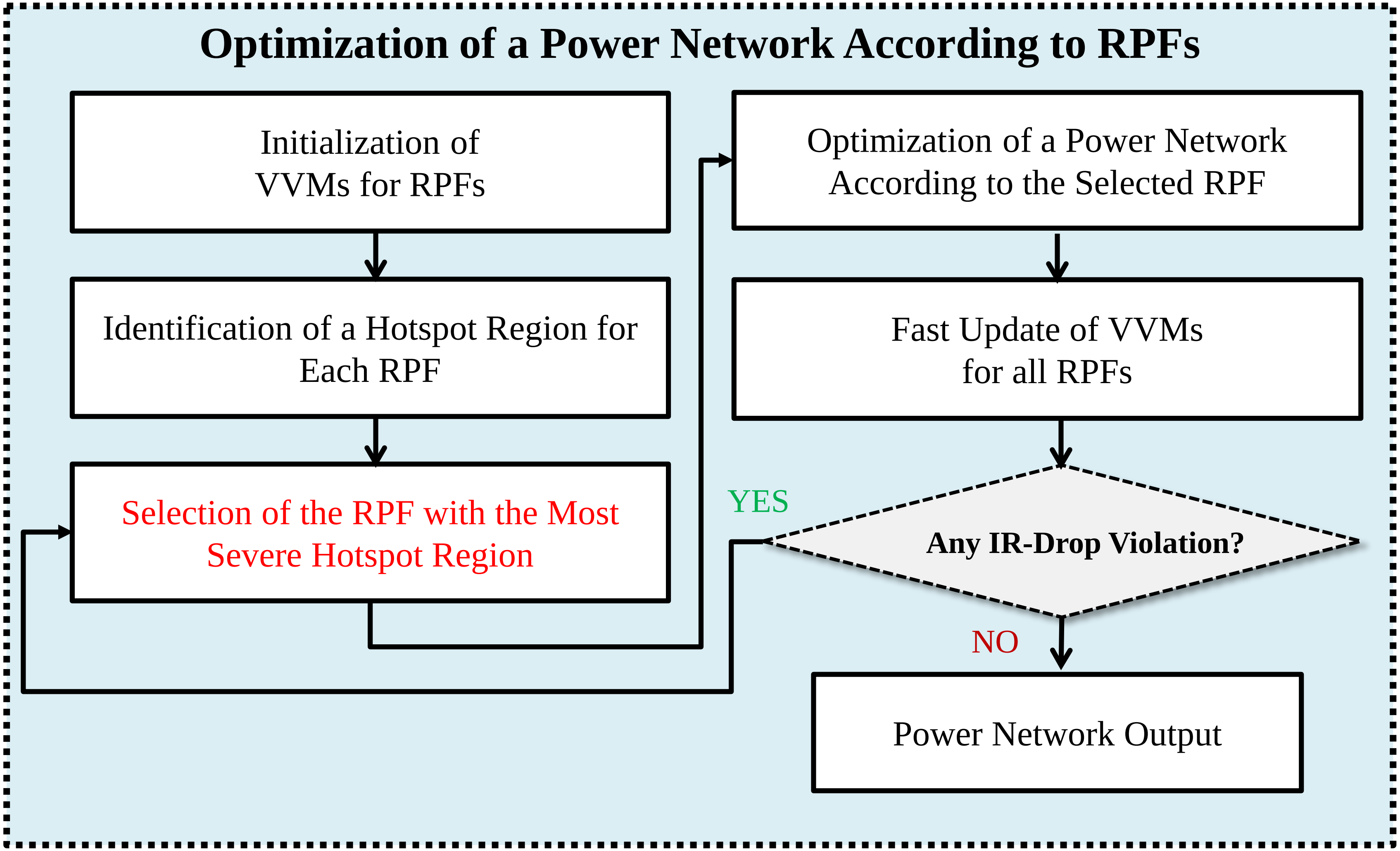
Identification of a Hotspot Region for Each RPF

- Discover the grid with the most serious IR-drop violation for each RPF.
- Generate a window with a proper size to cover the grid.



(a) RPF 1 Hotspot Region (b) RPF 2 Hotspot Region (c) RPF 3 Hotspot Region (d) RPF 4 Hotspot Region

Selection of RPF with the Most Serious IR-drop Region



Selection of RPF with the Most Severe IR-drop Region

- Select a RPF to optimize power network according to the hotspot regions in different RPFs f_i's.
- Use the following function to determine the priority of a f_i:

$$\Psi(f_i) = \sum_{g=1}^{h} \varphi_g * v_g^i$$

  - v_g^i denotes IR-drop violation value of grid g in a RPF f_i.
  - $\varphi_g = \begin{cases} 1, & \sigma \leq 1 \\ 10, & 1 < \sigma \leq 3 \\ 100, & \sigma > 3 \end{cases}$, violation grid g repeatedly appears in the window of σ RPFs.
  - The red region means violation repeatedly in the different RPFs.

0.23	0.62	0.45	0.29	0.14
0.32	0.68	0.52	0.21	0.54
0.29	0.44	0.61	0.24	0.40
0.19	0.18	0.38	0.30	0.39
0.16	0.34	0.24	0.13	0.37

(a) IR-drop Violation Region of RPF1

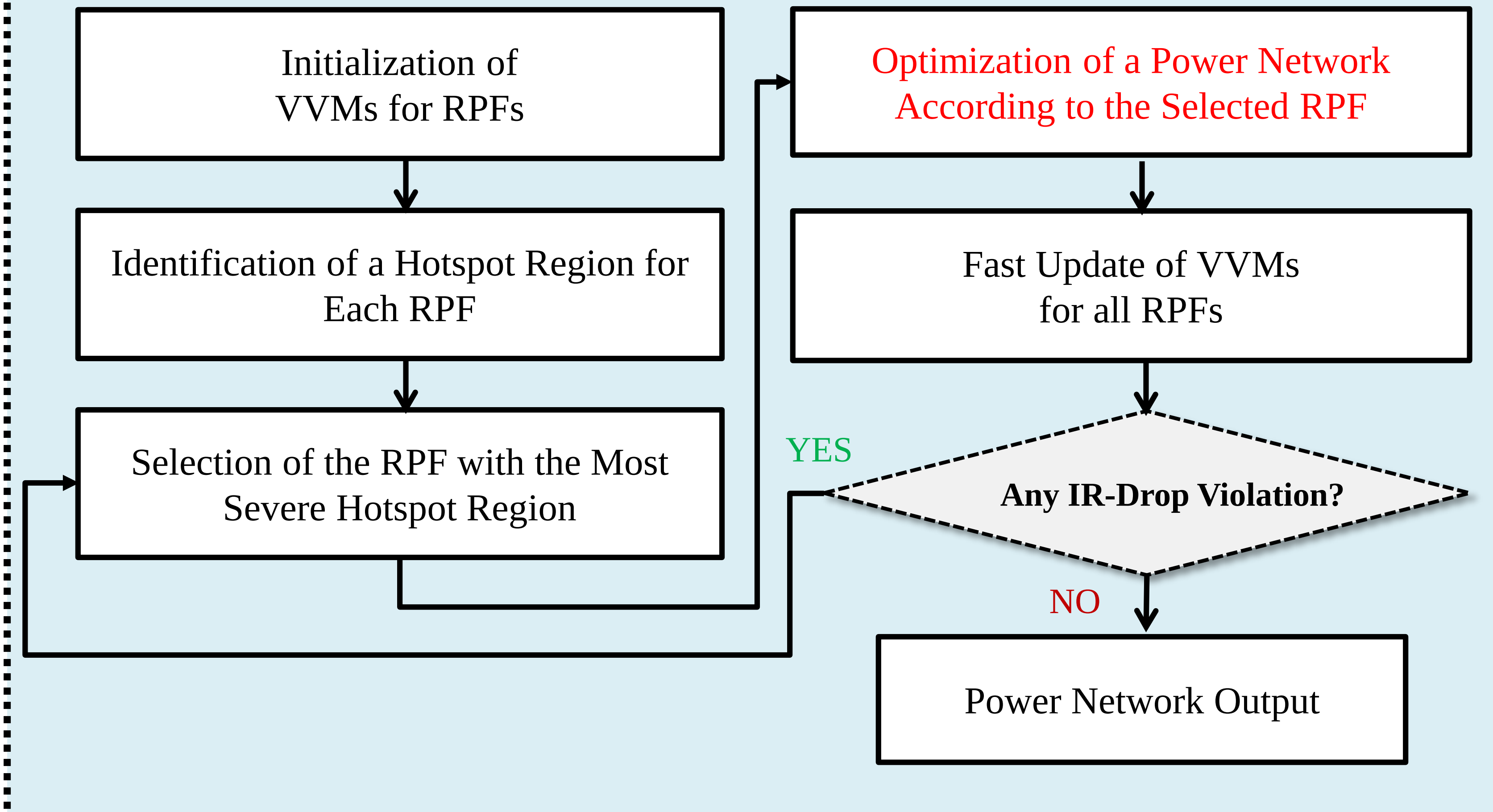
0.23	0.30	0.15	0.15	0.10
0.30	0.32	0.23	0.23	0.20
0.30	0.44	0.59	0.34	0.50
0.18	0.18	0.75	0.30	0.45
0.15	0.38	0.44	0.32	0.23

(b) IR-drop Violation Region of RPF2

0.23	0.22	0.15	0.51	0.21
0.40	0.31	0.38	0.61	0.42
0.21	0.34	0.62	0.81	0.71
0.41	0.11	0.44	0.37	0.76
0.49	0.23	0.27	0.14	0.13

(c) IR-drop Violation Region of RPF3

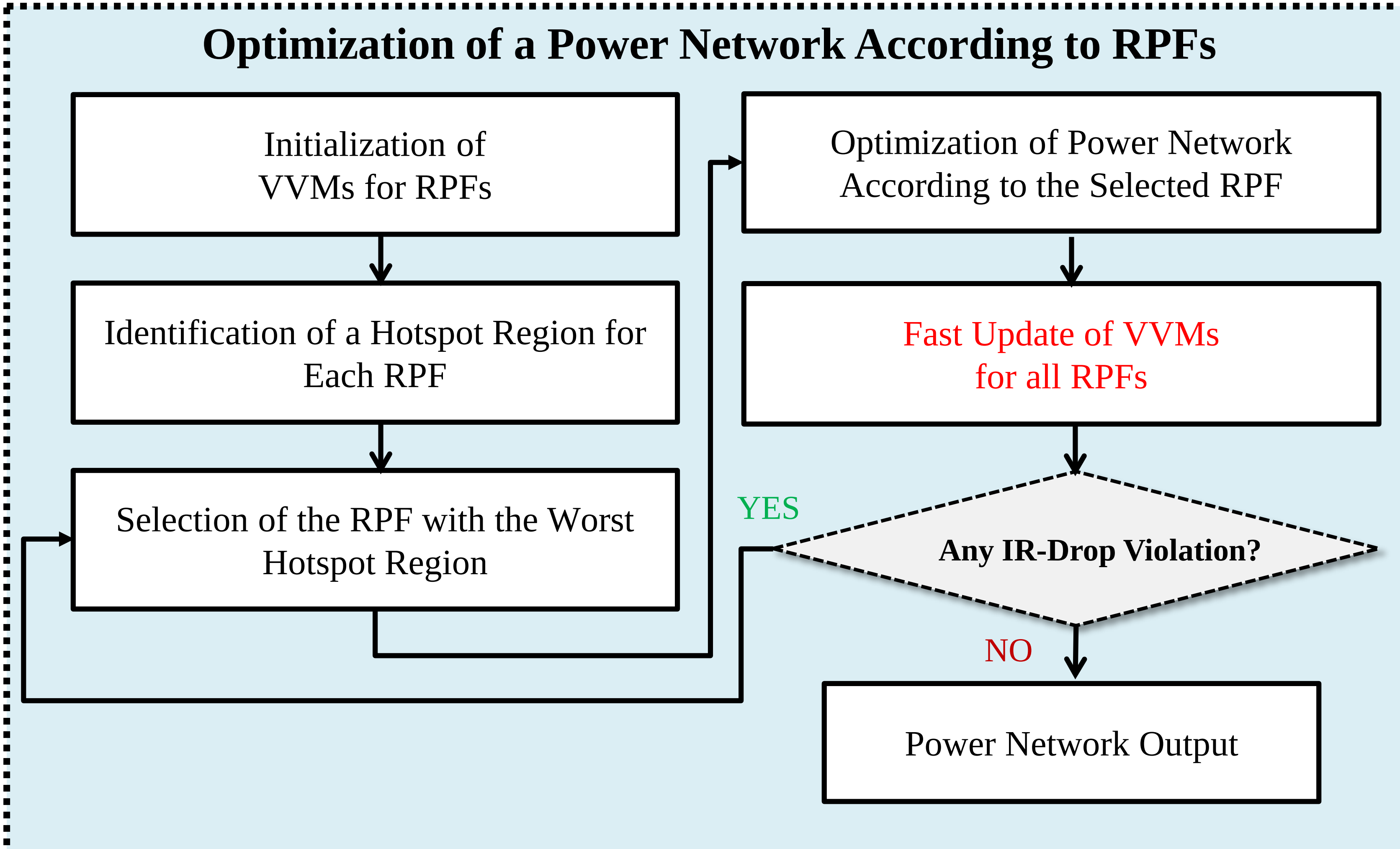
Optimization of a Power Network According to RPFs



VPSs Sizing

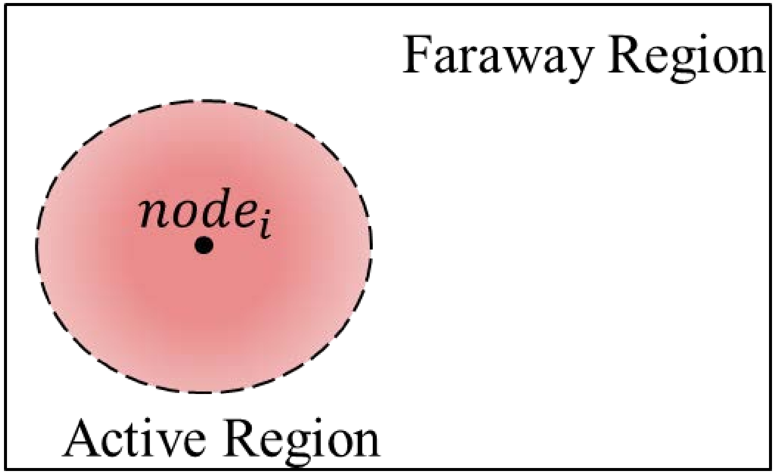
- Propose an iterative method to adjust the widths and lengths of VPSs in a region as follows:
  - Compute the equivalent VPS resistance of the regionwith the worst voltage violation, and compute the IR-drop violation ratio for each power stripe in the region.
  - Fix the width of each VPS to a larger ESW (effective stripe width) with respect to the current width ESW in the table constructed in advance.
  - Adjust length of VPSs according to its equivalent resistance and IR-drop ratio.

0.23	0.30	0.15	0.15	0.10
0.30	0.32	0.23	0.23	0.20
0.30	0.44	0.61	0.34	0.50
0.18	0.18	0.38	0.30	0.45
0.15	0.38	0.44	0.32	0.23
	S_0	S_1	S_2	


Optimization of a Power Network According to RPFs
Initialization of VVMs for RPFs
Identification of a Hotspot Region for Each RPF
Selection of the RPF with the Worst Hotspot Region
Optimization of Power Network According to the Selected RPF
Fast Update of VVMs for all RPFs
Any IR-Drop Violation?
YES
NO
Power Network Output

Fast Voltage Update

- Apply an efficient node based approach to update voltages of those nodes which are close to $node_i$ if we increase the VPS area at $node_i$.
  - It is quite time consuming to update the voltage of the entire power network.
  - The computation can be limited to a small region because the change at $node_i$ will propagate out and vanish after some distance.



Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Experimental Results

- Environments:

Programming Language		C++
Linux Workstation	CPU	Intel® Xeon® E5500 2.27GHz
	Memory	90GB
	System	Cent OS 5.1

- Our benchmarks are based on real designs from Himax.

Cir.	# of Cells	# of Mac.	Supply Voltage (V)	Static Power (W)	Max. Power (W)
Cir1	672952	222	1.10	0.033	0.0789
Cir2	193688	277	0.81	0.503	0.7802
Cir3	5909306	874	1.10	0.903	1.3270

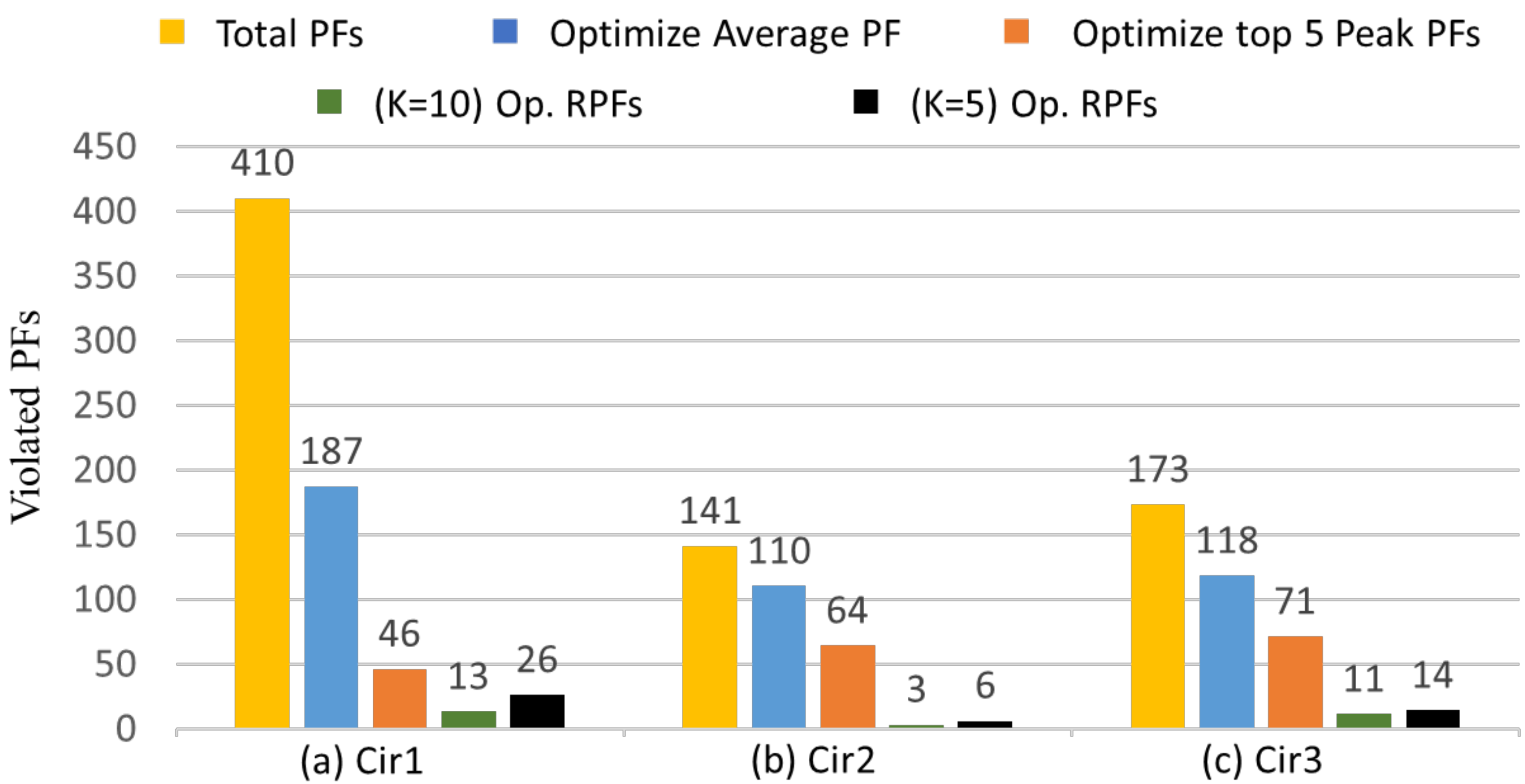
Comparison of Methods to Pick RPFs

- Compare with other approaches including average PF and top five worst PFs, and our K-clustering algorithm obtains the smallest "Total V.".
  - "Total V." denotes the ratios of violated PFs to the total PFs.
  - "Total V." by our approach are about 1/11 and 1/6 with respect to the average PF and the top five worst PF, respectively.

Circuit	Average PF Optimization				Top Five Peak PF Optimization				Selected RPF Optimization (K = 10)						Selected RPF Optimization (K = 5)					
	Area (10^6 μm^2)	Max IR drop (mV)	Total V. (%)	Op. (s)	Area (10^6 μm^2)	Max IR drop (mV)	Total V. (%)	Op. (s)	Area (10^6 μm^2)	Max IR drop (mV)	Total V. (%)	Cluster (s)	Op. (s)	Total (s)	Area (10^6 μm^2)	Max IR drop (mV)	Total V. (%)	Cluster (s)	Op. (s)	Total (s)
Cir1	4.85	68.4	45.7	29	4.94	64.1	11.2	70	4.93	63.4	3.1	215	56	271	4.91	63.6	6.3	235	32	267
Cir2	25.9	49.7	78.0	48	27.8	47.3	45.3	234	27.6	46.6	2.3	190	197	387	26.6	46.3	4.2	212	140	352
Cir3	89.2	65.1	68.2	753	93.5	62.7	43.9	2672	92.2	59.3	6.3	490	2388	2878	91.9	60.1	8.1	540	1954	2494
Nor.	0.97	1.07	11.4	0.54	1.02	1.02	5.9	1.72	1.01	0.99	0.60	0.90	1.46	1.09	1.00	1.00	1.00	1.00	1.00	1.00

Comparison of Methods to Pick RPFs(cont'd)

- The numbers of PFs which violate IR-drop constraint in different approaches for each circuit.



Comparison of a Different Clustering Algorithm

- Compare K-clustering with K-means, which is the most classic clustering algorithm.
  - Modify K-clustering which does not need to specify number of clusters K in advance.
- Compare two algorithms using different cost functions.

$$diff(i,j) = \frac{1}{h}\sum_{b=0}^{h}\left|x_b^i - x_b^j\right|$$

Algorithm	K-means						K-clustering					
Cost	Diff.			Dist.			Diff.			Dist.		
	Area (10^6 um^2)	Total Vio.(%)	Total Time(s)	Area (10^6 um^2)	Total Vio.(%)	Total Time(s)	Area (10^6 um^2)	Total Vio.(%)	Total Time(s)	Area (10^6 um^2)	Total Vio.(%)	Total Time(s)
Cir1	4.92	4.7	583	4.92	4.3	621	4.93	3.2	291	4.93	3.1	282
Cir2	26.2	5.3	527	26.3	5.1	652	27.8	2.5	457	27.4	2.5	474
Cir3	92.0	7.6	3568	92.0	6.8	3915	92.4	5.8	3374	92.9	5.4	3219
Nor.	0.98	16.8	1.43	0.98	1.59	1.50	1.003	1.035	1.014	1.00	1.00	1.00

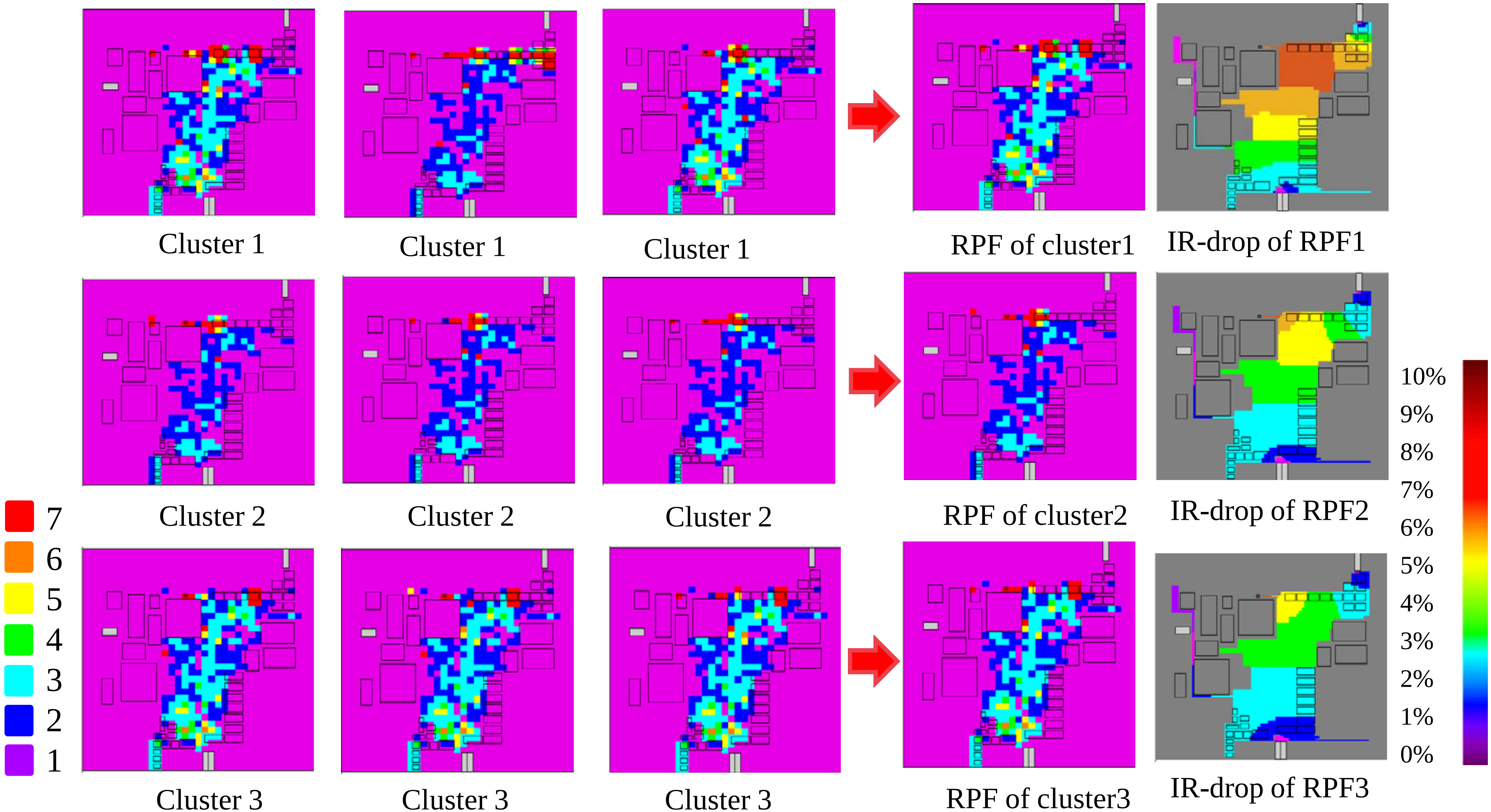
Comparison of Our Sizing Approach with Other Approaches

- Compare our window based sizing algorithm with other approach including iterative methodology and sequential linear programming methodology (SLP for short) [1].

Circuit	Original Status			Iterative method				SLP method				Our method			
	Area (10^6 um^2)	Total OV.	Max. IR drop (mV)	Area(10^6 um^2)	Total OV.	Max. IR drop (mV)	Time (s)	Area (10^6 um^2)	Total OV.	Max. IR drop (mV)	Time (s)	Area (10^6 um^2)	Total OV.	Max. IR drop (mV)	Time (s)
Cir1	4.59	398	67.1	5.53	470	62.5	7012	4.98	440	62.9	387	4.93	423	63.4	282
Cir2	24.9	8536	52.6	35.8	40549	45.8	5221	29.1	8762	46.1	721	27.4	8682	46.7	474
Cir3	86.5	10267	66.9	137.2	14683	57.9	36850	102.6	11067	58.2	4255	92.9	10845	59.0	3219
Nor.	0.92	0.96	1.11	1.30	1.23	0.98	15.77	1.06	1.02	0.99	1.40	1.00	1.00	1.00	1.00

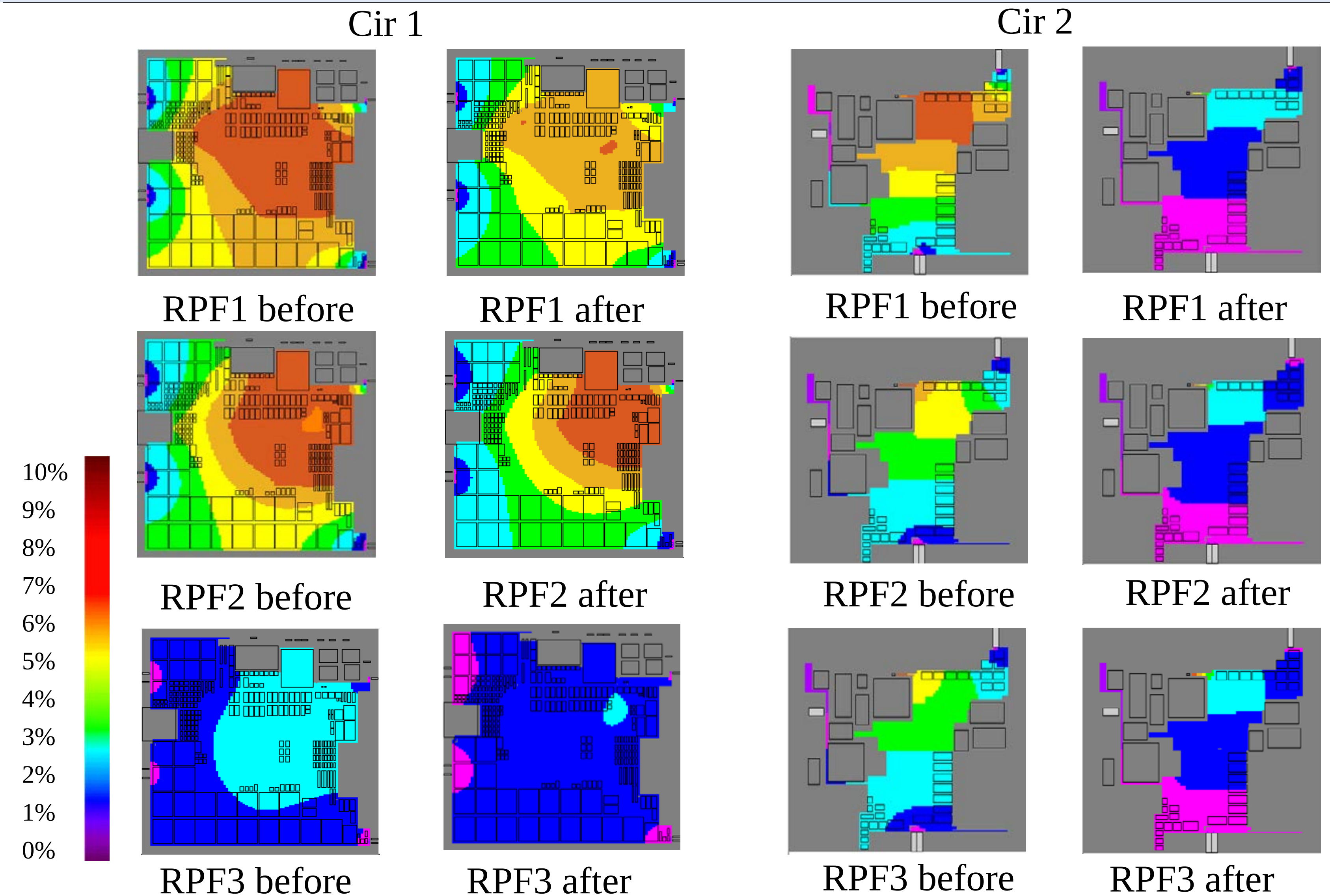
[1] S.S.-Y. Liu, C.-J. Lee, C.-C. Huang, H.-M. Chen, C.-T. Lin and C.-H. Lee, "Effective Power Network Prototyping Via Statistical-based Clustering and Sequential Linear Programming," in Proc. DATE, pp. 1701-1706, Mar. 2013.

Experimental Results



Cir 2

Experimental Results



Outline

- Introduction
- Problem Formulation
- Overview of Our Methodology
- Discovery of Representative Power Consumption Files
- Power Network Optimization Algorithm
- Experimental Results
- Conclusion

Conclusion

- Propose a power network optimization based on clustering algorithm for dynamic IR-drop.

- Propose an efficient method to resolve dynamic IR-drop violations for industrial cases.
  - Construct PDMs from a VCD file and classify them into several categories through a clustering algorithm.
  - Find a representative power consumption file in each cluster.
  - Optimize power network by resizing the VPSs in the most severe voltage violation region in each RPF.

- The experimental results have demonstrated that our approach can greatly reduce the ratio of violated power profiles to total critical power profiles compared to intuitive approaches.

Thank You For Your Attention