Network Simplex Method Based Multiple Voltage Scheduling in Power-Efficient High-Level Synthesis

Cong Hao, Song Chen, Takeshi Yoshimura

Graduate School of IPS, Waseda University, Japan

01/23/2013

Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Introduction – MVS

- Low-power scheduling
  - Reducing supply voltage is always efficient
  - **Multiple Voltage Scheduling (MVS)** is promising
- In **MVS**, resources with different supply voltages are available
  - **Higher supply voltage:** larger power, shorter delay
  - **Lower supply voltage:** smaller power, longer delay
- **MVS**: to find voltage assignment and scheduling for operations

Previous Works

- Scheduling Techniques for Variable Voltage Low Power Designs *(Y. R. Lin, C. T. Hwang, A. C.-H.Wu, DAES 1997)*
  - **ILP formulation** for latency and resource constrained MVS
  - **An iterative improvement heuristic** was proposed
- Modified force-directed scheduling for peak and average power optimization using multiple supply-voltages *(A. Allam, ICICDT 2006)*
  - Formulation for latency-constrained MVS
  - **Two phase to independently for power and resource**
- Scheduling with Integer Time Budgeting for Low-Power Optimization *(W.jiang, ASPDAC 2008)*
  - **LP formulation** for latency-constrained MVS
  - **A time-budgeting-based heuristic** to for power and resource

Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power and Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Problem Definition of MVS

- **Input** of **Latency-constraint** MVS
  - **A Data Flow Graph** (DFG)
  - **Latency Constraint T_{con}** (Max. allowable control step)
  - **Possible delays** for each operation
    - Adder: {1, 2, 3}; Multiplier: {2, 3, 4}
- **Goal**
  - Find (1) a **delay(voltage) assignment** and
    (2) an **operation scheduling**
  - which minimize **power consumption** and **resource usage** simultaneously as

$$\sum_{v \in V} Power(v) + \alpha \sum Cost(Res)$$

Approaches to MVS Problem

Two *sub-problems* of Multiple Voltage Scheduling

= Delay Assignment (Power Minimization) + Operation Scheduling (Resource Minimization)

- **Simultaneous** Optimization
  - **Desirable** yet **time-consuming** and **complicated**

    Simultaneous "**Delay Assignment**" and "**Scheduling**"

- **Independent** Optimization
  - Solve delay assignment followed by scheduling

    ① **Delay Assignment** → ② **Scheduling**

Approaches to MVS Problem

Two *sub-problems* of Multiple Voltage Scheduling

= Delay Assignment (Power Minimization) + Operation Scheduling (Resource Minimization)

- **Simultaneous** Optimization
  - **Desirable** yet **time-consuming** and **complicated**

Simultaneous "**Delay Assignment**" and "**Scheduling**"

- **Independent** Optimization
  - Solve delay assignment followed by scheduling

① **Delay Assignment** → ② **Scheduling**

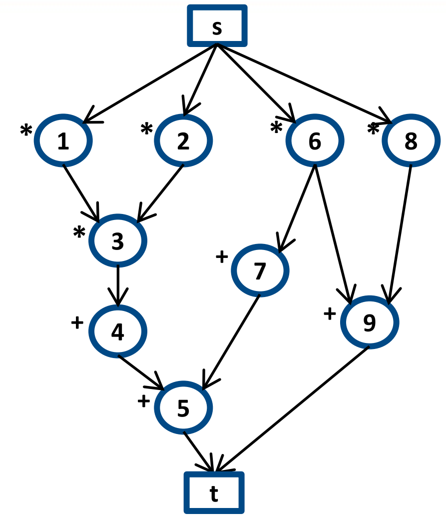
First consider

Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power and Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Power Optimization (ILP Formulation)

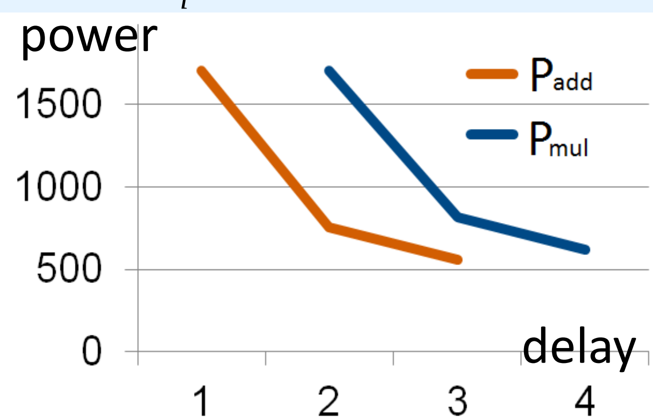
- **Given**: DFG G=(V,E)
- **Variable**: for each v_i
  - p_i is the starting control step of v_i
  - q_i is the ending control step of v_i



- **Constraints:**
  - **Data dependency**
    $$q_i - p_j \le 0, \forall e = (v_i, v_j) \in E$$
  - **Operation Delay**
    $$d_{\min} \le q_i - p_i \le d_{\max}, \forall v_i \in V$$
  - **Latency**
    $$0 \le t - s \le T_{con}$$

- **Objective Function**

$$\min : \sum_{v_i \in V} P_{type(v_i)} (q_i - p_i)$$



Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power and Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Linear Programming Relaxation

- The **constraint matrix** of ILP formulation is
  - The **incidence matrix** of a “**splitting graph**”
  - **Totally Unimodular**
  - ILP can be **relaxed to Linear Programming**

Data dependency

$$q_i - p_j \le 0, \forall (v_i, v_j) \in E$$

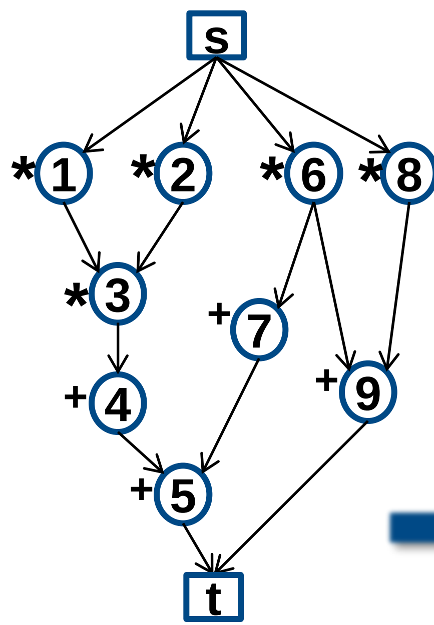
Operation Delay

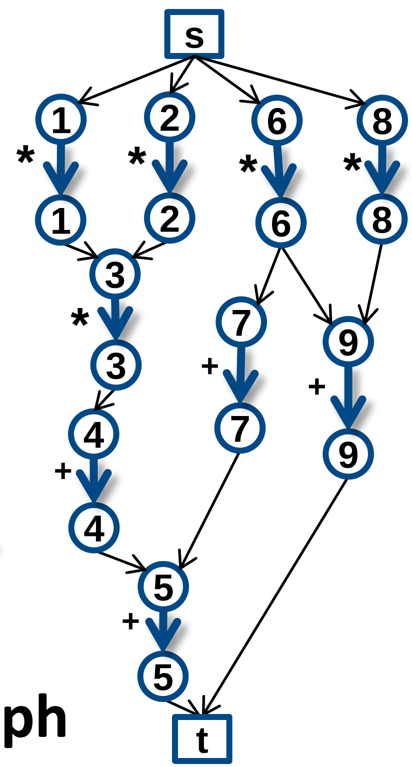
$$d_{\min} \le q_i - p_i \le d_{\max}, \forall v_i \in V$$

Latency

$$0 \le t - s \le T_{con}$$

Totally Unimoludar Matrix





Splitting Graph

Sub-optimality of Independent Approach

- **Independent** Optimization
  - Solve delay assignment followed by scheduling

① **Delay Assignment** ➡ ② **Scheduling**

Sub-optimality of Independent Approach

- **Independent** Optimization
  - Solve delay assignment followed by scheduling

① **Delay Assignment** ➡ ② **Scheduling**

Optimally solved
(Linear Programming)

Sub-optimality of Independent Approach

- **Independent** Optimization
  - Solve delay assignment followed by scheduling

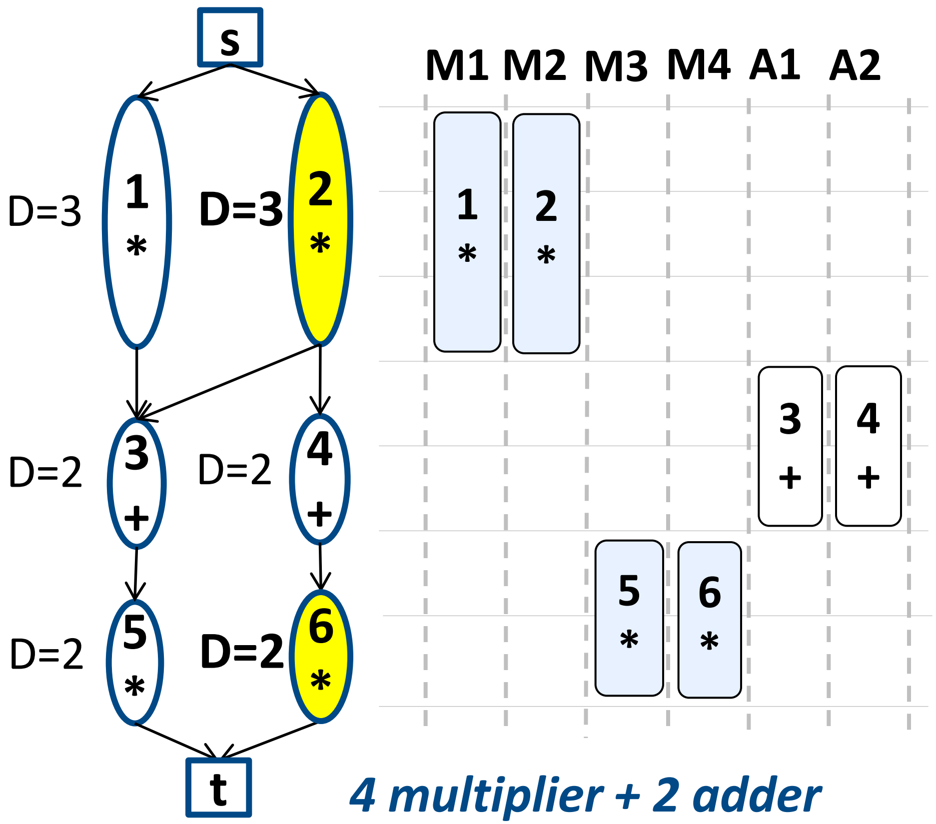
① Delay Assignment ➡ **② Scheduling**

Optimally solved
(Linear Programming)

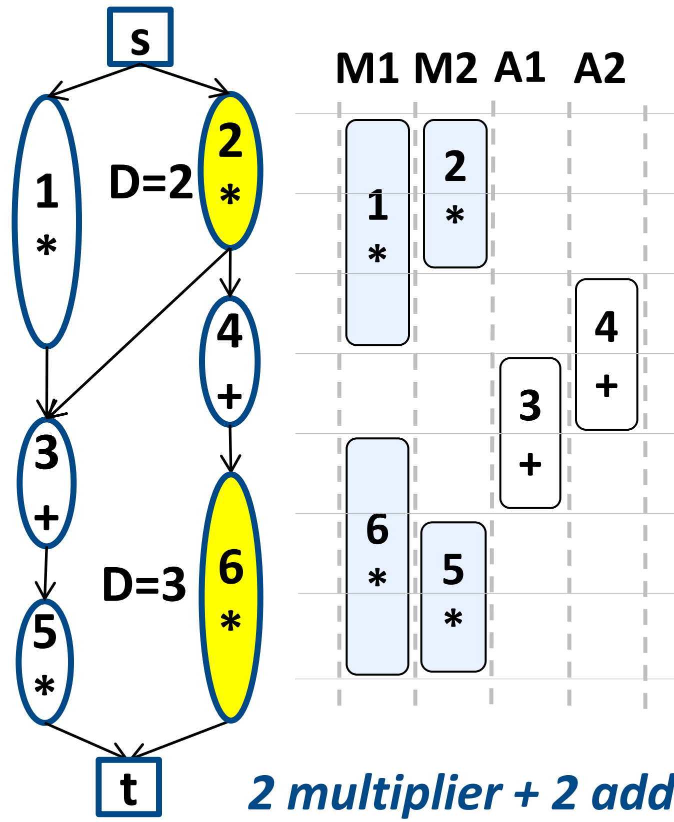
- **Delay Assignment** can be solved **optimally**
- However, Independently optimizing power and resource may **result in sub-optimal solutions**

Sub-optimality of Independent Approach

- Power are both optimized but resource…



4 multiplier + 2 adder



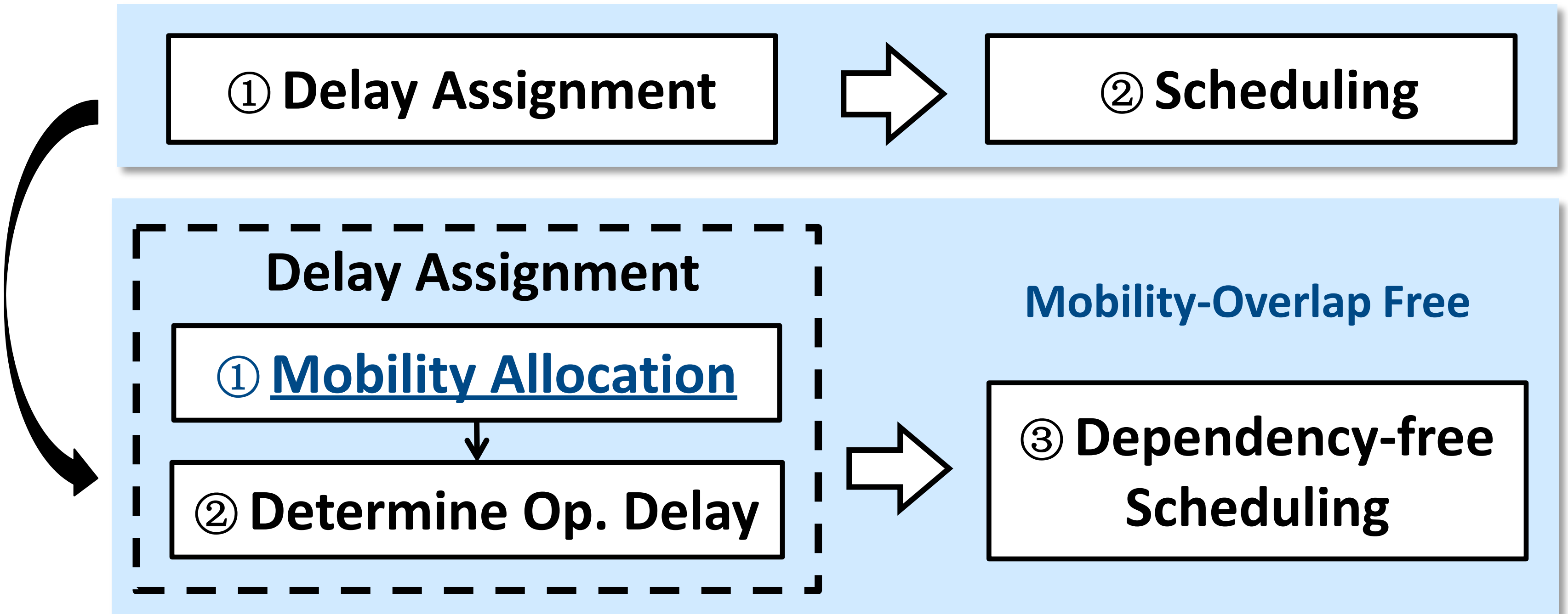
2 multiplier + 2 adder

Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Semi-Simultaneous Proposal

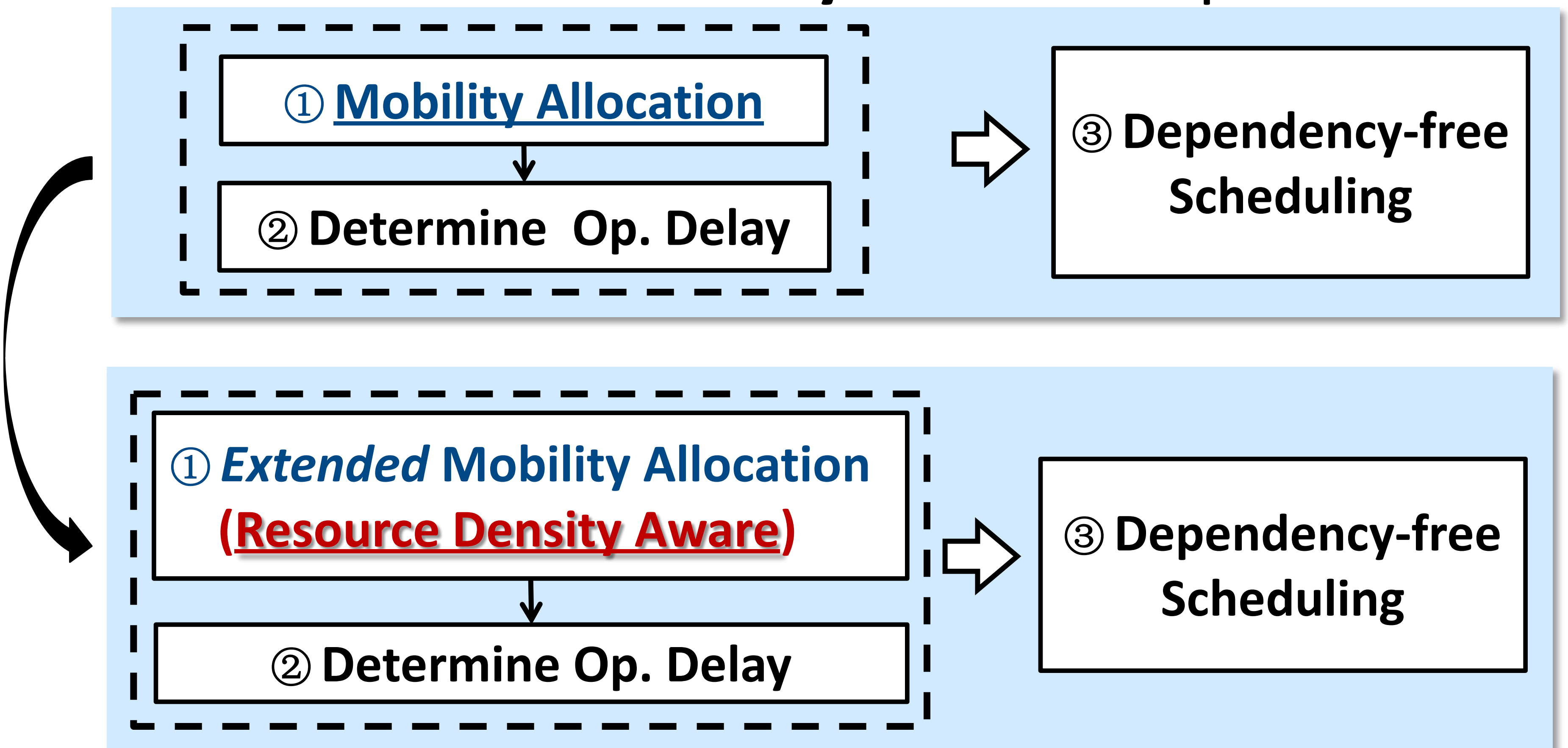
- **Apply resource-aware power optimization**
  - First introduce **mobility** for operations
  - Transfer **delay assignment** to **mobility allocation**



Power Minimization **Resource Minimization**

Resource-Aware Power Optimization

- Further **extend mobility allocation** problem



Stage 1:
Extended Power Minimization

Stage 2:
Resource Minimization

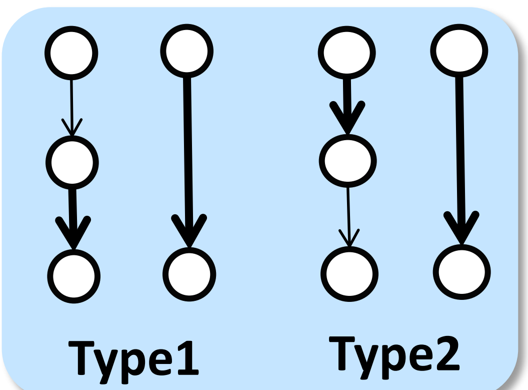
Outline

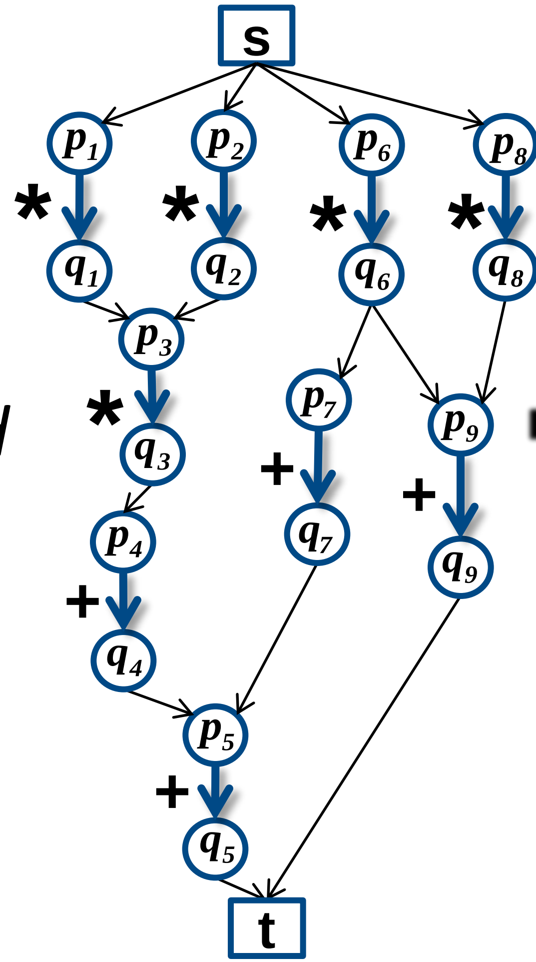
- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Mobility & Mobility Graph

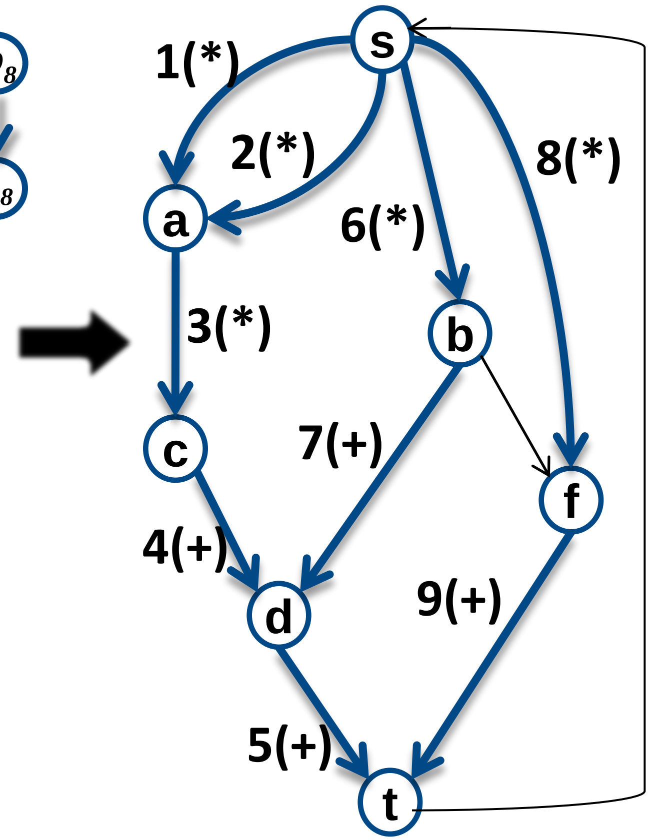
- **Mobility**

  An interval of consecutive control steps an operation could be scheduled to





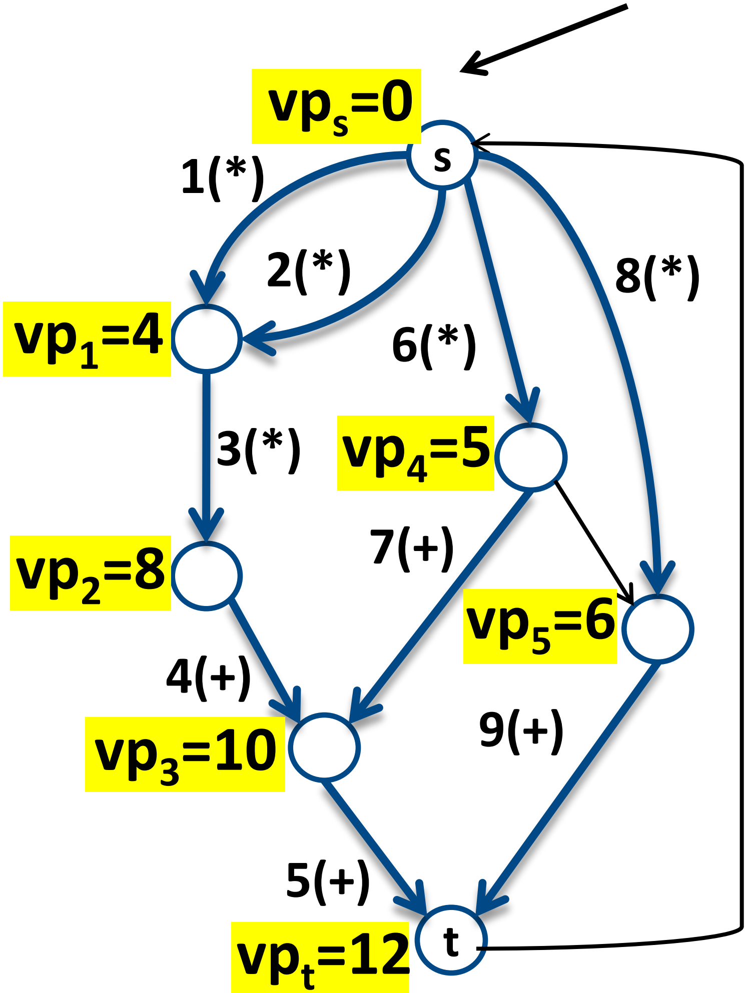
Splitting Graph
"zero mobility"

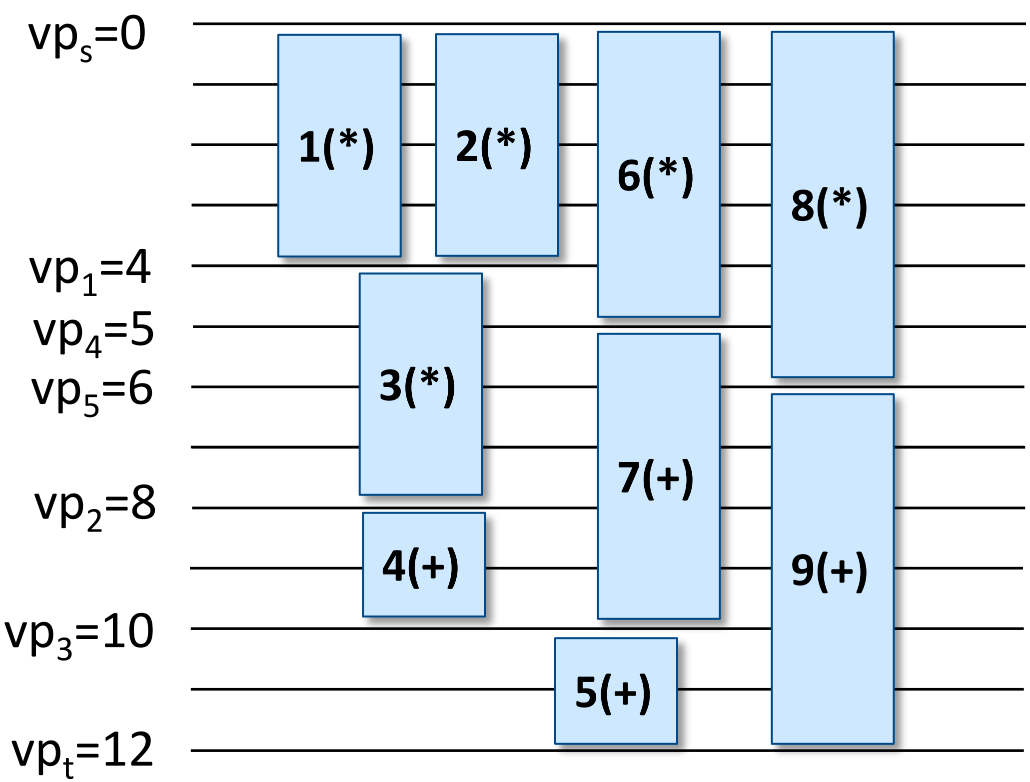


Mobility Graph

Mobility Allocation by Vertex Potential

Each vertex is associated with a ***"vertex potential" -- vp_i***

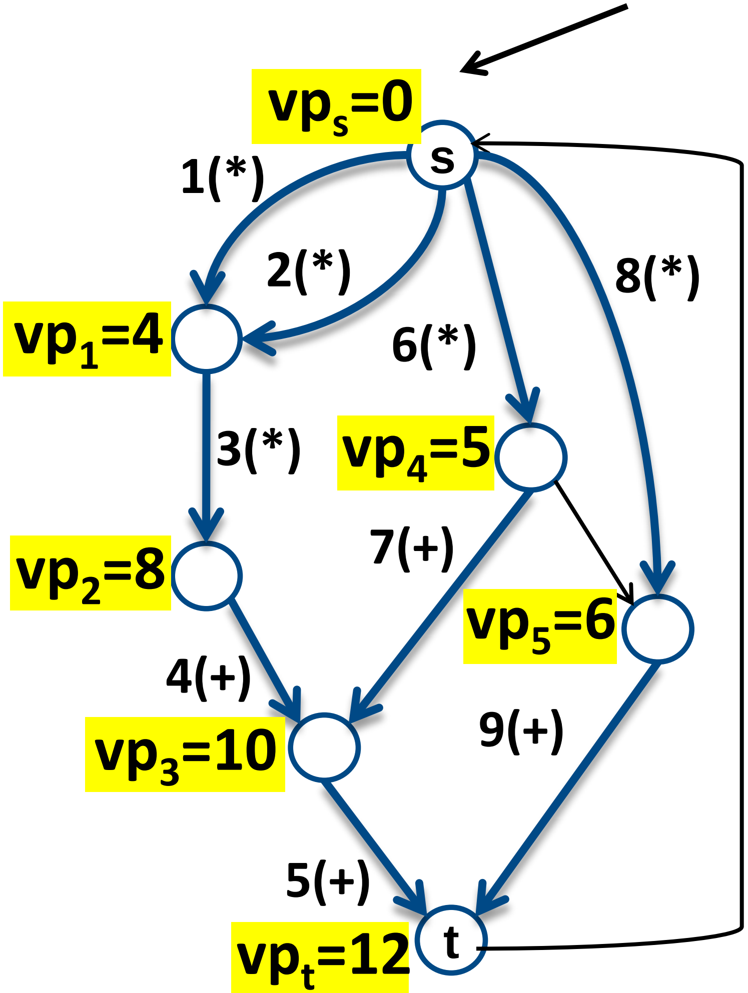


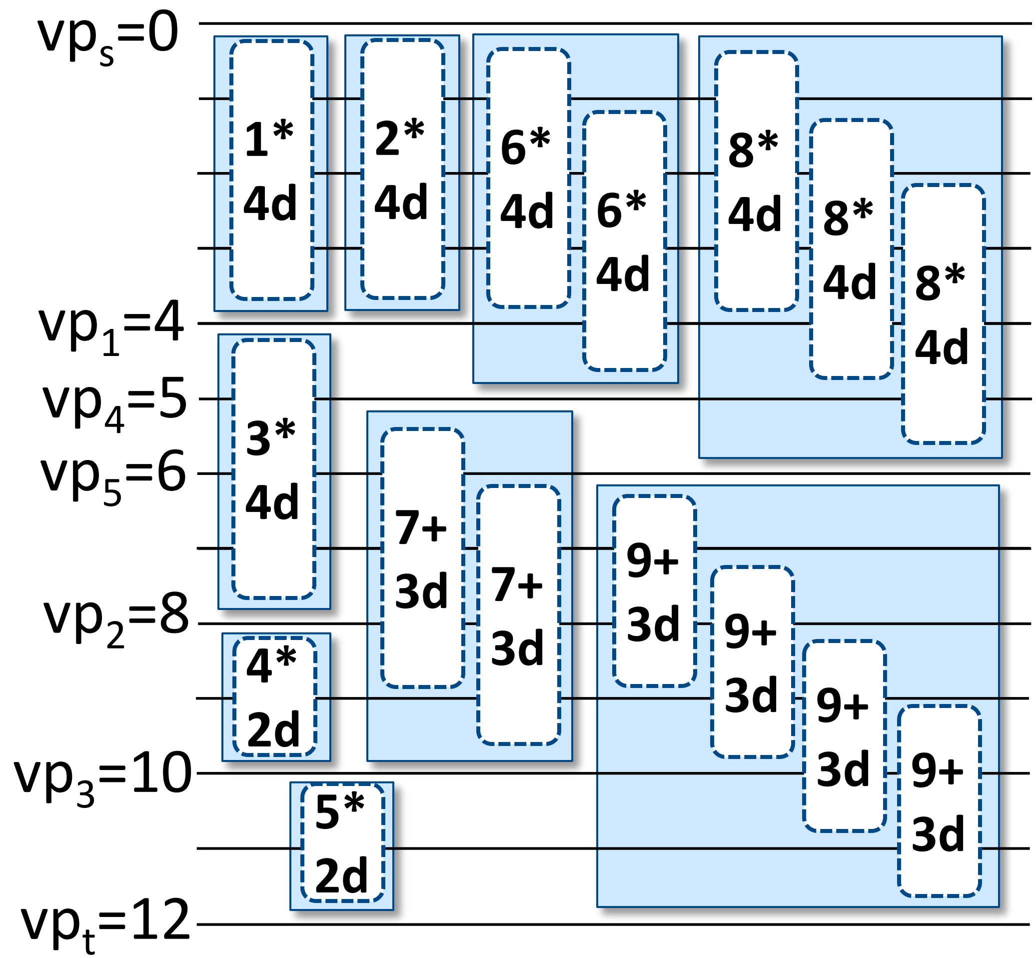


Mobility-overlap-free: Dependency-free

Mobility Allocation by Vertex Potential

*Each vertex is associated with a **"vertex potential" -- vp_i***





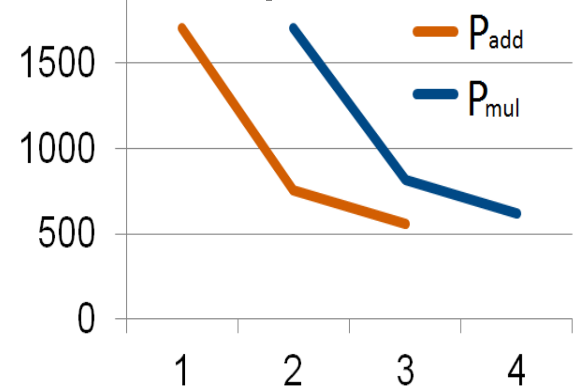
Mobility Allocation Problem

- **Variables**
  - **Vertex Potential (VP)** of vertices
- **Constraints**
  - Mobility Graph $G_m = (V_m, E_m)$
- **Objective Function**

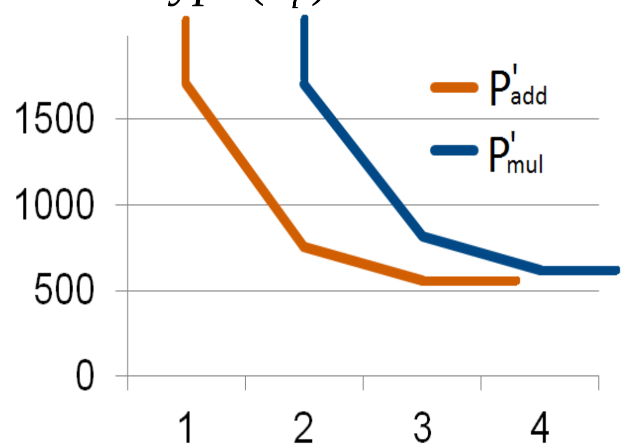
$$\min : \sum_{v_i \in V} P_{type(v_i)}(q_i - p_i)$$

$$\min : \sum_{e=(v_i, v_j)} P'_{type(e)}(vp_j - vp_i)$$

$P_{type(v_i)}$



$P'_{type(e_i)}$

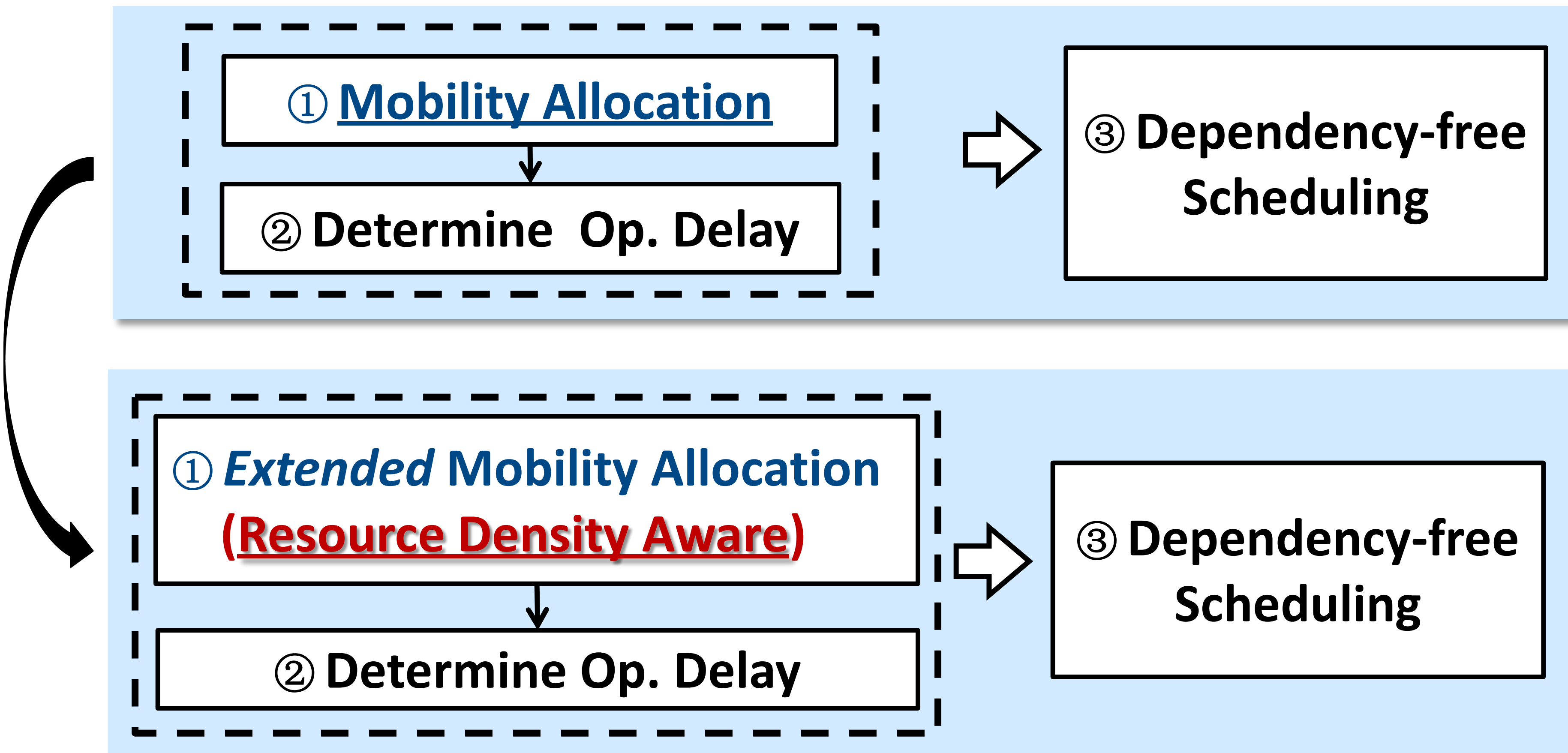


Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

(Recall)Resource-Aware Power Optimization

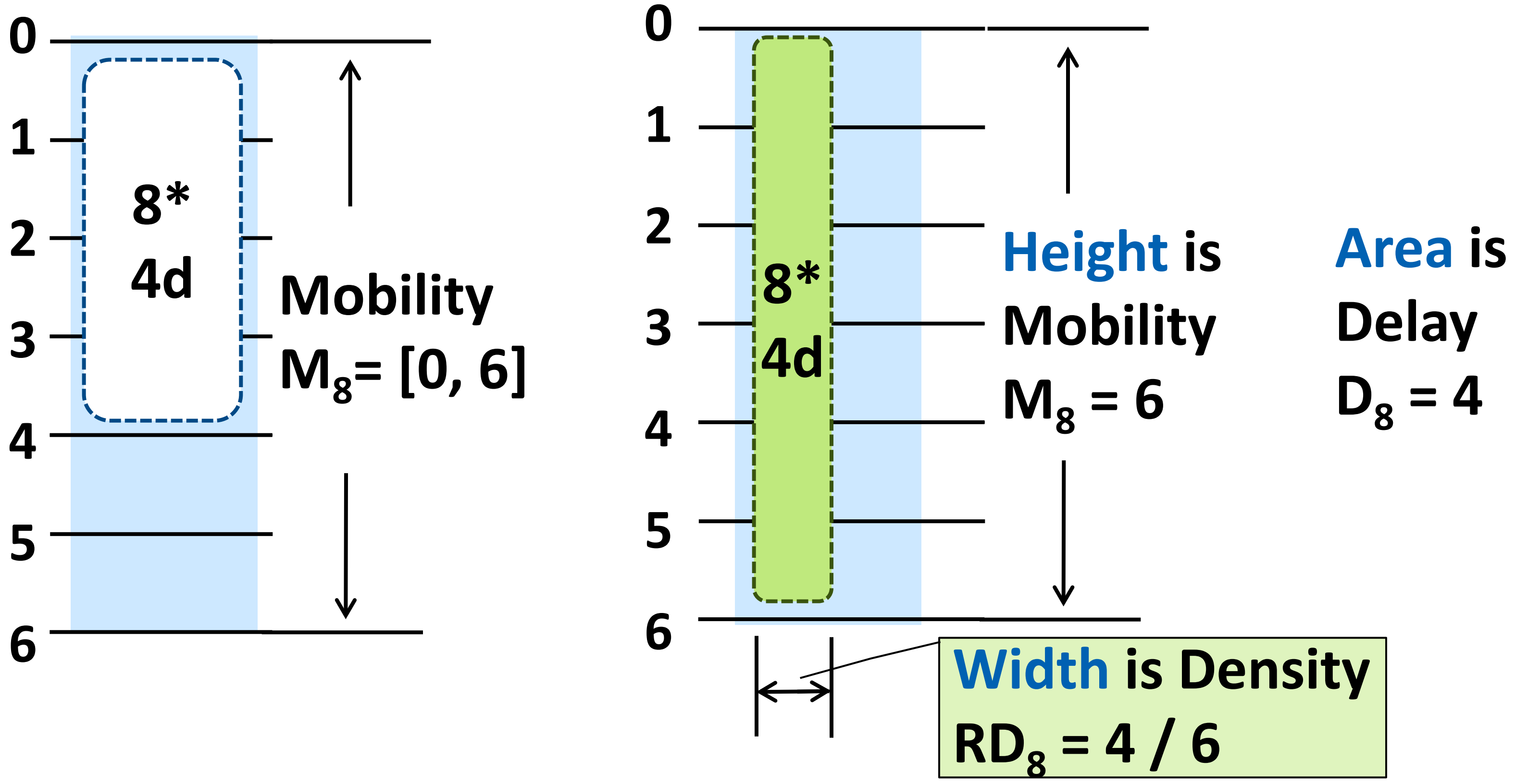
- Further **extend mobility allocation** problem



Extended Power Minimization **Resource Minimization**

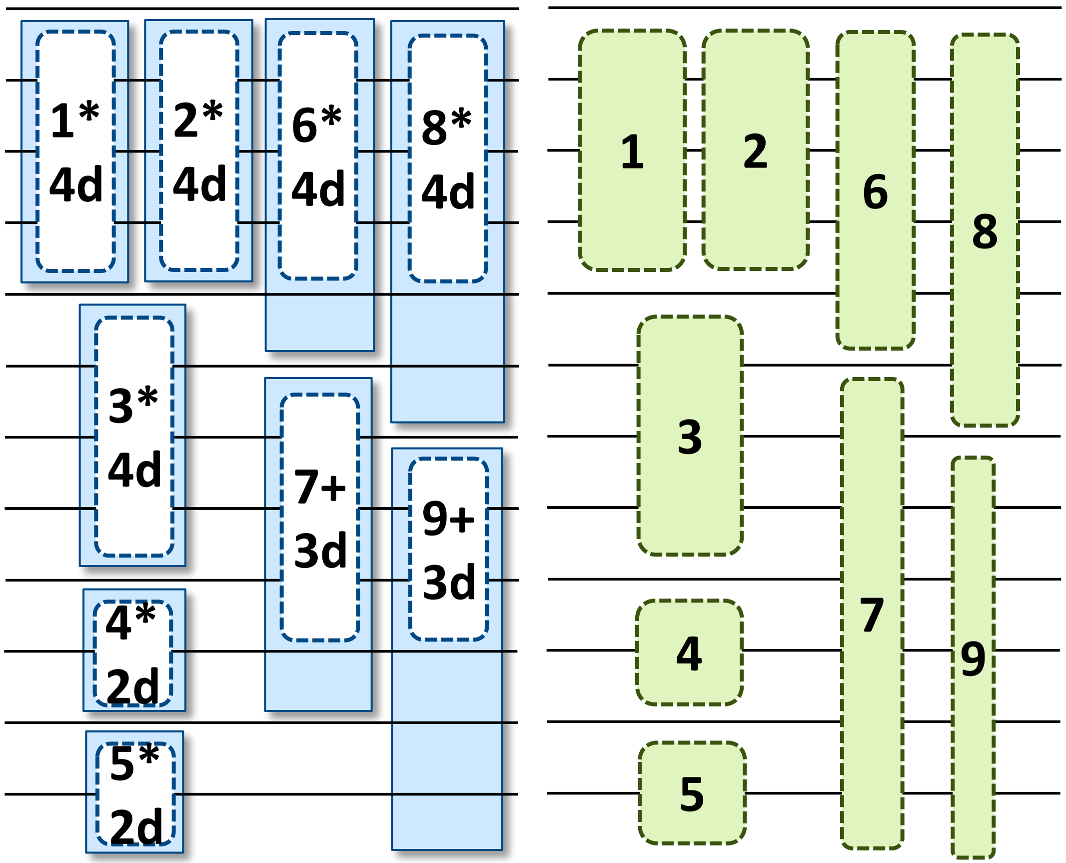
Resource Density

- **Resource density** of each operation is computed from **mobility allocation** result
- For an operation v_i , $RD(v_i) = dly(v_i) / |M(v_i)|$



Balancing Resource Density

- **Resource density of each time slot** is expected to **distribute normally** **(Force Directed)**

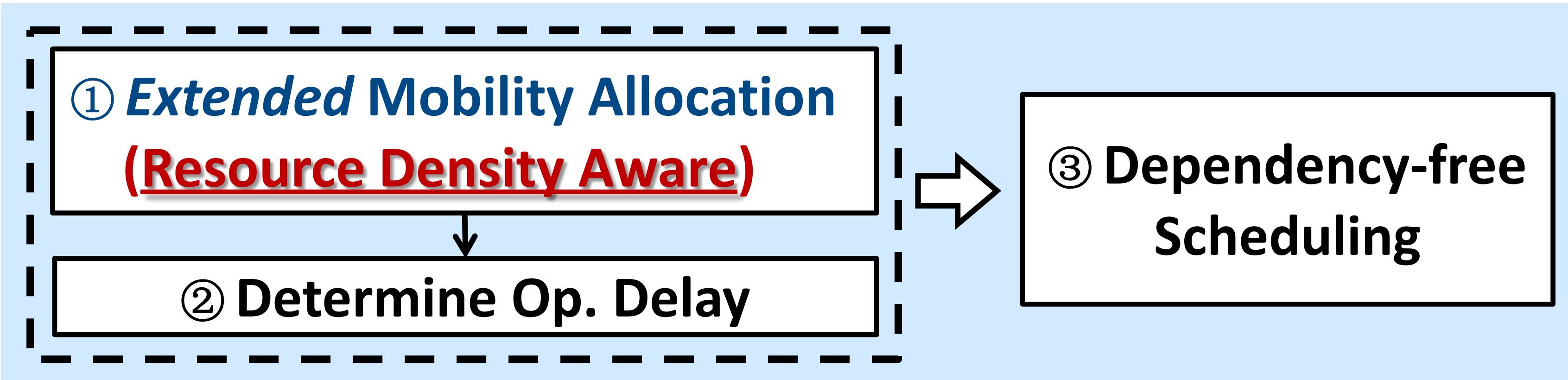


- To minimize **density variation**

$$\sum_{ts}(den_{ts} - den_{avg})^2$$

A function of **Vertex Potential**

Extended Power Optimization

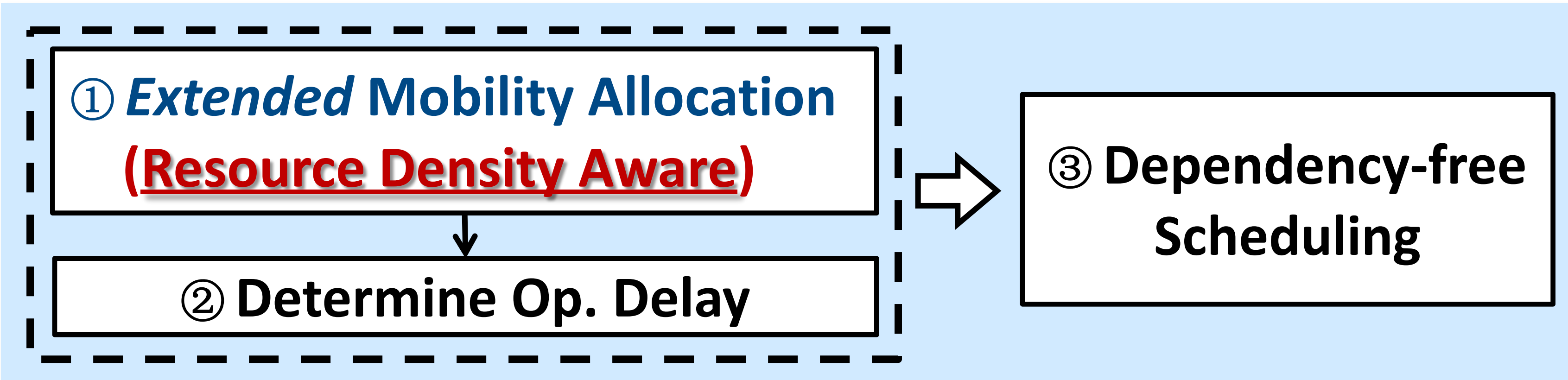


- Extended Objective Function

$$\min: \sum_{e=(v_i, v_j)} P'_{type(e)} (vp_j - vp_i)$$

Power Minimization

Extended Power Optimization



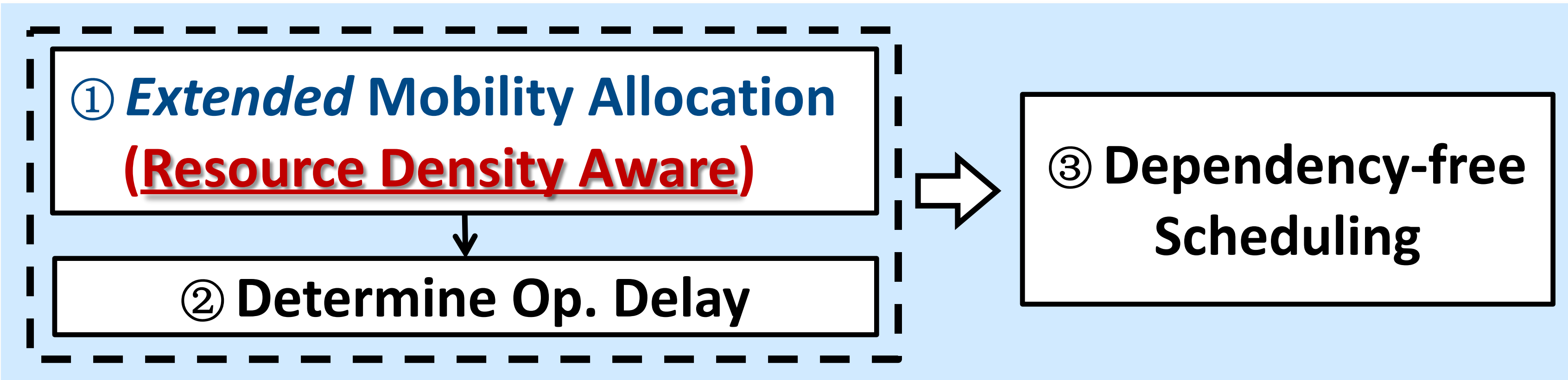
- Extended Objective Function

$$\min: \sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \alpha \cdot \sum_{ts} (den_{ts} - den_{avg})^2$$

Power Minimization

Density variation minimization

Extended Power Optimization



- Extended Objective Function

$$\min: \sum_{e=(v_i, v_j)} P'_{type(e)}(vp_j - vp_i) + \alpha \cdot \sum_{ts} (den_{ts} - den_{avg})^2$$

Power Minimization

Density variation minimization

- Both terms are **Function of Vertex Potential**

Solve Extended Power Optimization

- To minimize $\sum (den_{ts} - den_{avg})^2$
  - A solution ***vpt**$_i$* is expected to be obtained
  - We call it **target vertex potential (*VP**)**
- Objective function is further changed as

$$\min: \sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \alpha \cdot \sum_{ts} (den_{ts} - den_{avg})^2$$

$$\min: \sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \beta \cdot \sum_{v_i \in V_m} | vp_i - vpt_i |$$

Solve Extended Power Optimization

- To minimize $\sum (den_{ts} - den_{avg})^2$
  - A solution ***vpt_i*** is expected to be obtained
  - We call it **target vertex potential (*VP**)**
- Objective function is further changed as

$$\min : \sum_{e=(v_i, v_j)} P'_{type(e)} (vp_j - vp_i) + \alpha \cdot \sum_{ts} (den_{ts} - den_{avg})^2$$

$$\min : \sum_{e=(v_i, v_j)} P'_{type(e)} (vp_j - vp_i) + \beta \cdot \sum_{v_i \in V_m} | vp_i - vpt_i |$$

Obtained from minimizing $\sum (den_{ts} - den_{avg})^2$

Solve Extended Power Optimization

- To minimize $\sum (den_{ts} - den_{avg})^2$
  - A solution ***vpt**$_i$* is expected to be obtained
  - We call it **target vertex potential (*VP**)**

- Objective function is further changed as

$$\min: \sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \alpha \cdot \sum_{ts} (den_{ts} - den_{avg})^2$$

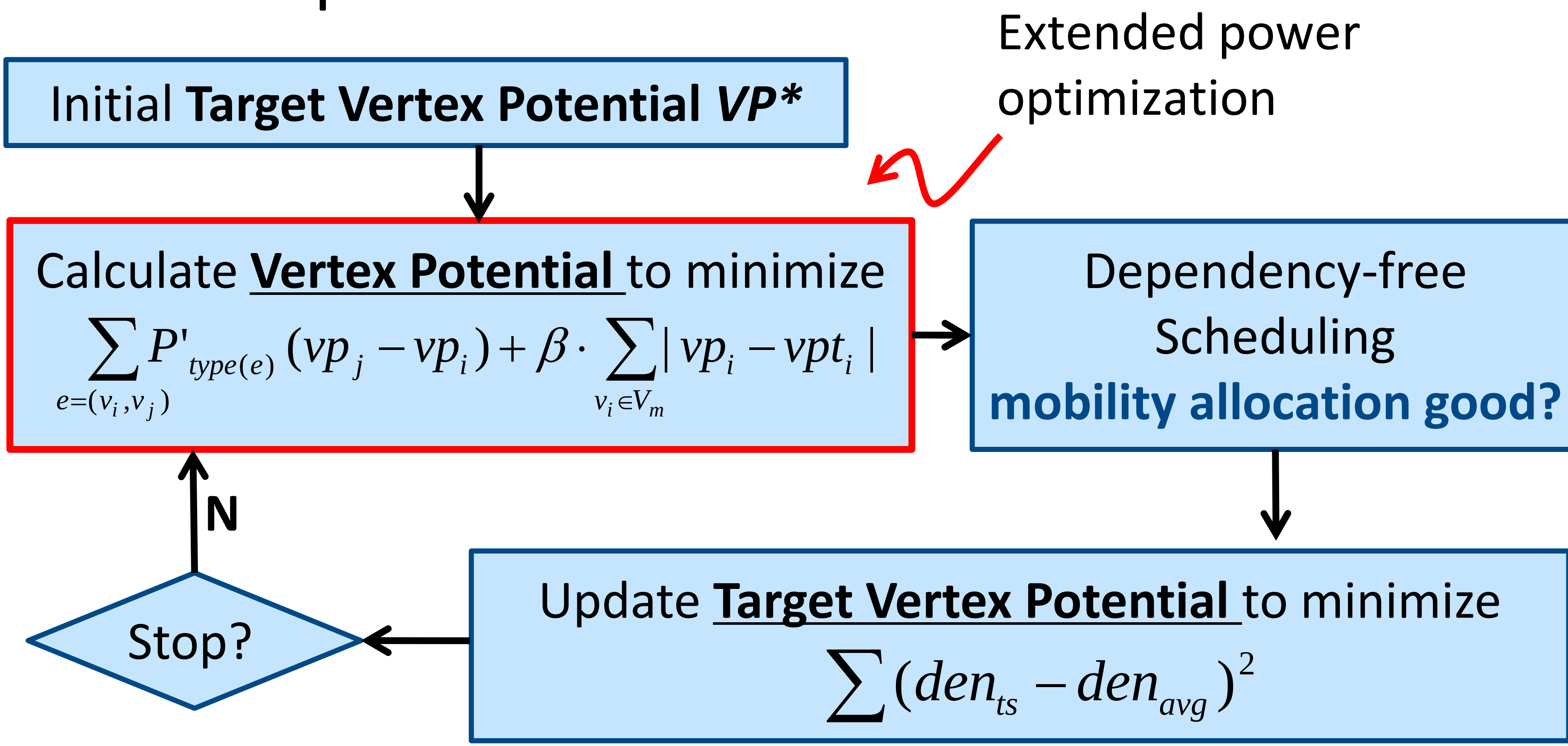
⬇

$$\min: \sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \beta \cdot \sum_{v_i \in V_m} | vp_i - vpt_i |$$

- The new objective function is piecewise **linear programming**

Solve Extended Power Optimization

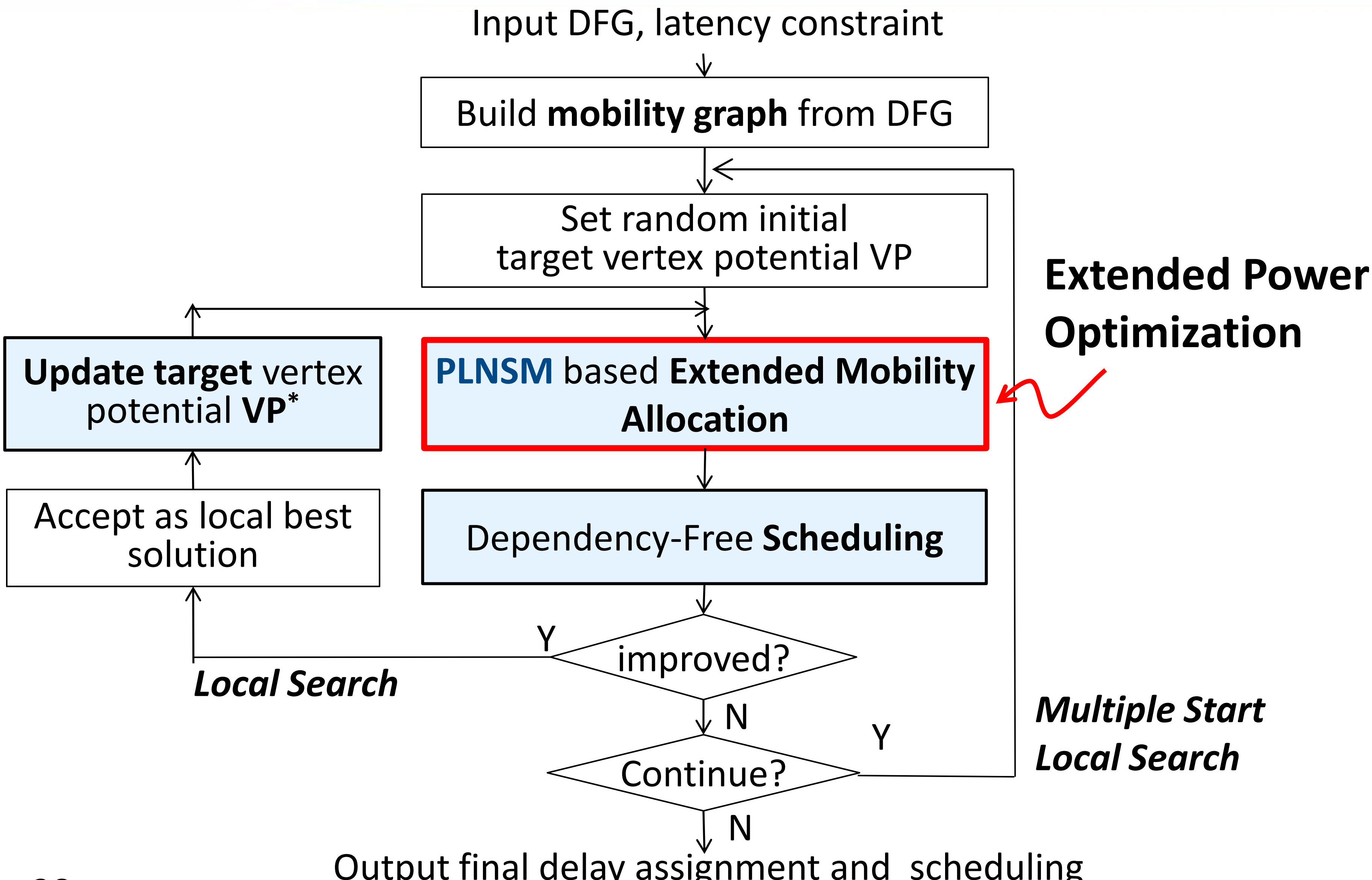
- An **iterative** proposal to struggle for a good vertex potential solution



Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Flow-Chart of Proposed NPMVS

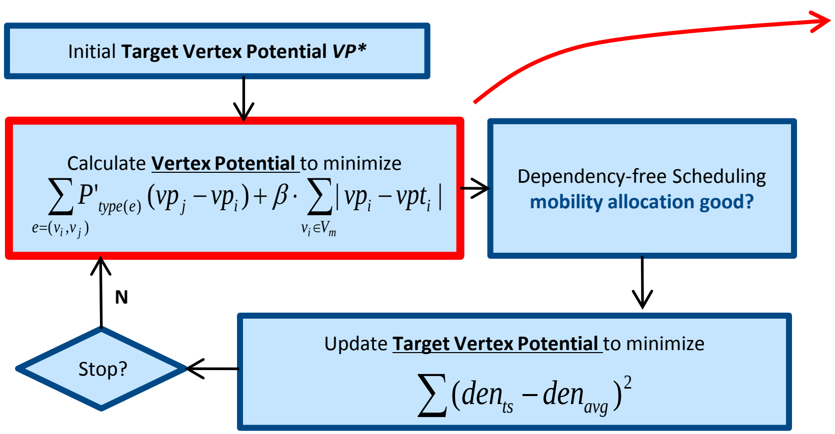


Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Specific Solver for Power Optimization

- **Extended Power Optimization** is crucial and performed **repeatedly**
  - **High-efficiency** is expected
  - A specific solver **PLNSM** is proposed
  - **P**iecewise-**L**inear extended **N**etwork **S**implex **M**ethod



Calculate **Vertex Potential** to minimize

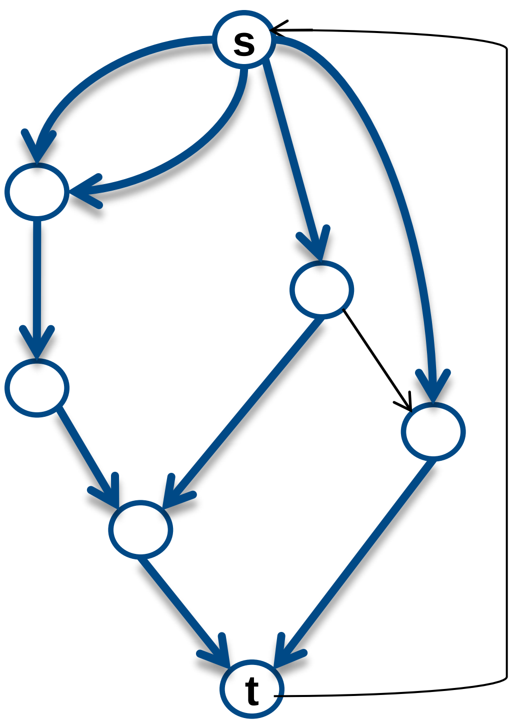
$$\sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \beta \cdot \sum_{v_i \in V_m} |vp_i - vpt_i|$$

- **Extended Power Optimization**
- **Solved by the PLNSM repeatedly**

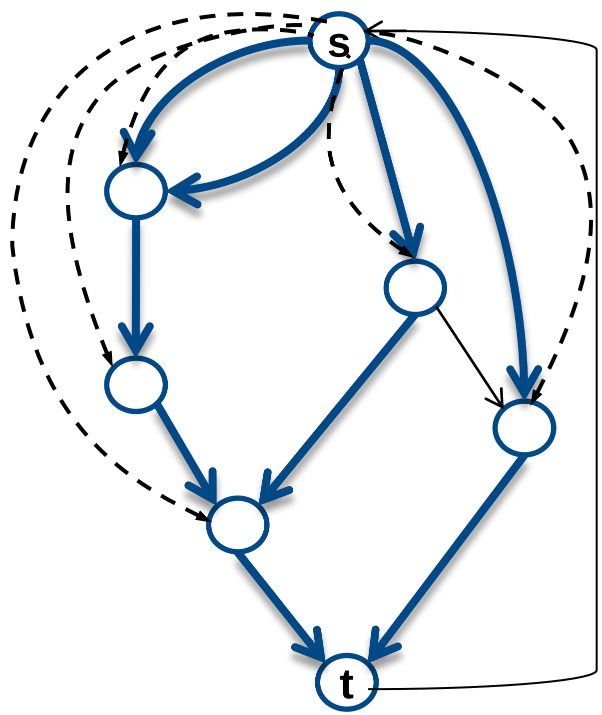
Objective Function with Target Potential

Formulation for extended power optimization is still ***Totally Unimodular*** *& can be relaxed to* ***Linear Programming****!*

$$\sum_{e=(v_i,v_j)} P'_{type(e)}(vp_j - vp_i) + \beta \cdot \sum_{v_i \in V_m} | vp_i - vpt_i |$$



Mobility graph



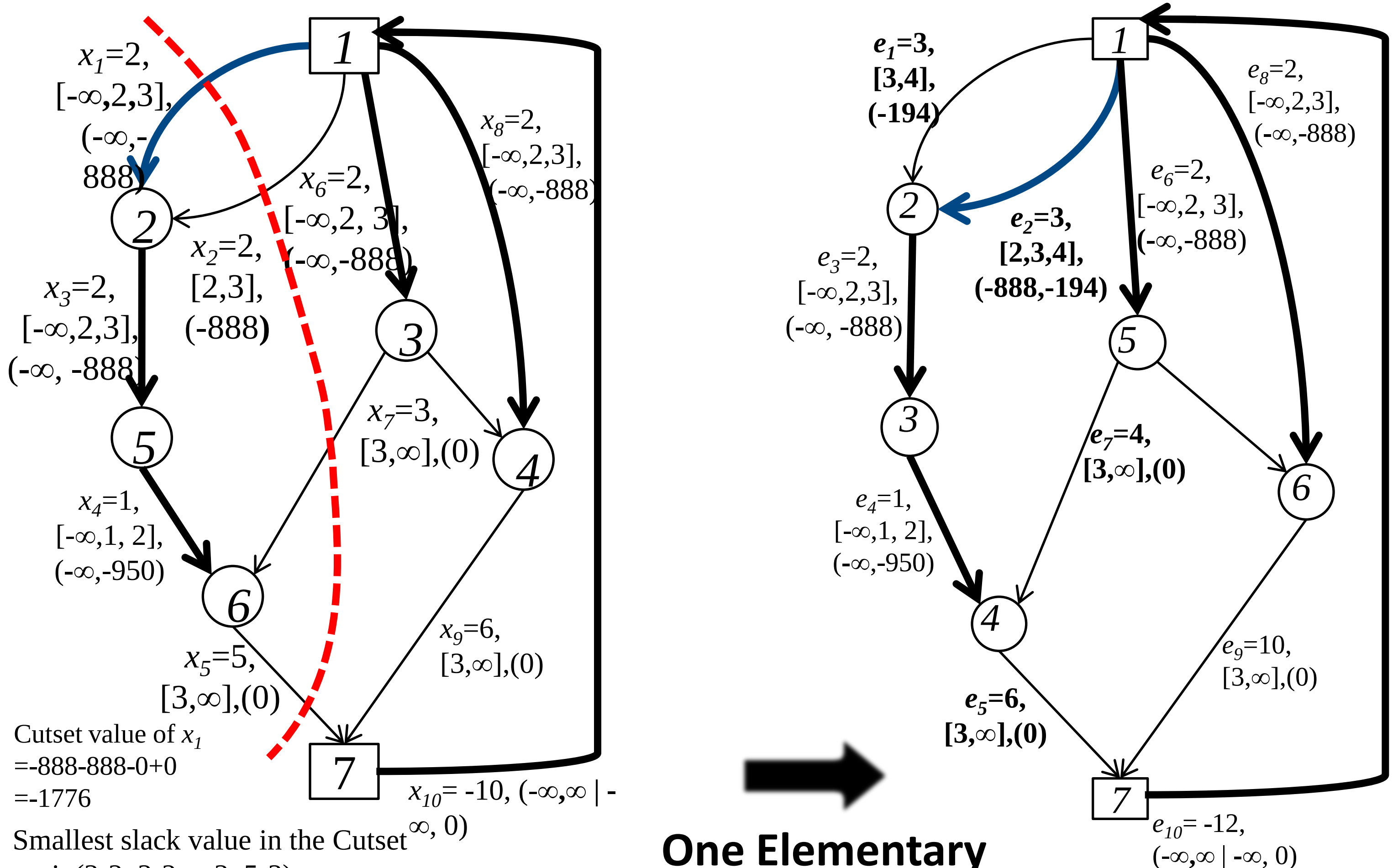
Extended mobility graph

The PLNSM Solver

- **Linear Programming** formulation
  - Allow us to adopt Network Simplex Method
- **Network Simplex Method (NSM)**
  - Works more efficient than **Simplex Method** under **network graphs (Our formulation)**
  - High-efficiency when **performed repeatedly**
- When dealing with **piecewise-linear functions**
  - Simplex Method introduces several times additional variables
  - **NSM** can easily be extended to **PLNSM** without any additional variables or constraints

Network Simplex Method (NSM)

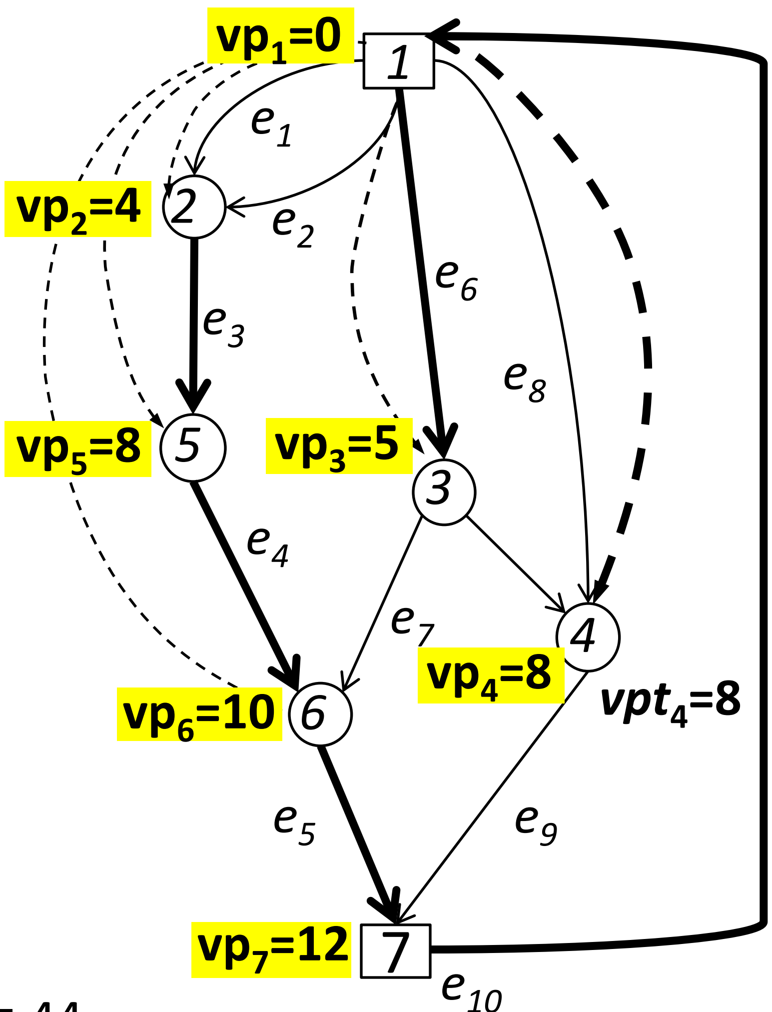
- Basic idea – perform **tree transformation**

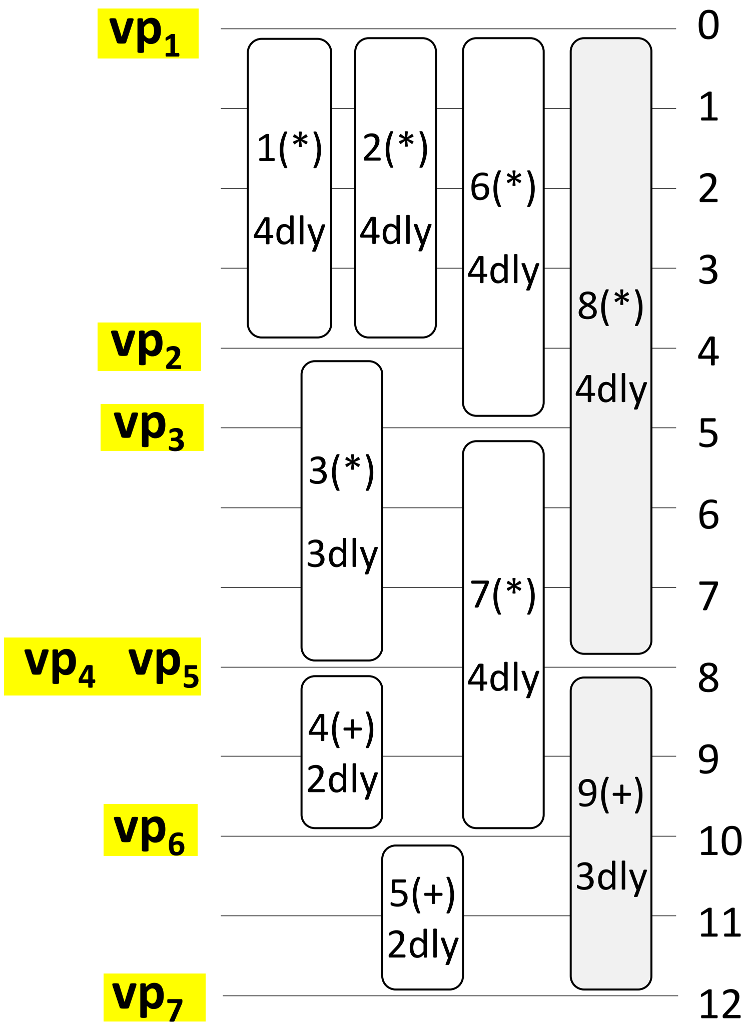


Smallest slack value in the Cutset
=min(3-2, 3-2,∞-3, 5-3)
=1 ← transformation between (x_1, x_2)

Network Simplex Method (NSM)

- Final tree & mobility allocation result





Outline

- Introduction & Previous Work
- Formulation for Multiple Voltage Scheduling
  - ILP formulation for Power Optimization
  - Linear Programming Relaxation
- Proposal for Power **and** Resource Optimization
  - Mobility Allocation
  - Extended Power Optimization
  - Proposed Multiple Voltage Scheduling Algorithm
  - Specific PLNSM Solver
- Experimental Results & Conclusion

Efficiency of PLNSM Solver

- Environment: Linux, AMD Opteron 2.6GHz & 4GBMem
- *DFG* shows the graph size (|*V*|,|*E*|) of **DFG**
  G_m shows the graph size (|*V*|,|*E*|) of **mobility graph**
- **PLNSM** (Network Simplex Method with Piecewise-Linear extension) achieves **80X+** speedup than *lp_solver**

	DFG	G_m	LP(ms)	PLNSM(ms)	cmp(X)
AD2	47,110	46,61	12.330	0.128	96.3
AE	54,143	49,69	17.550	0.160	109.7
AR	29,42	18,29	3.550	0.044	80.7
DIFF	9,16	7,9	0.787	0.012	65.6
ELLIP	35,67	30,35	7.020	0.070	100.3
MPEG	54,114	41,60	16.400	0.180	91.1
FFT	134,234	122,201	37.877	1.768	21.4
AVR					80.7

lp_solver: a general LP solving package from http://lpsolve.sourceforge.net/5.5/

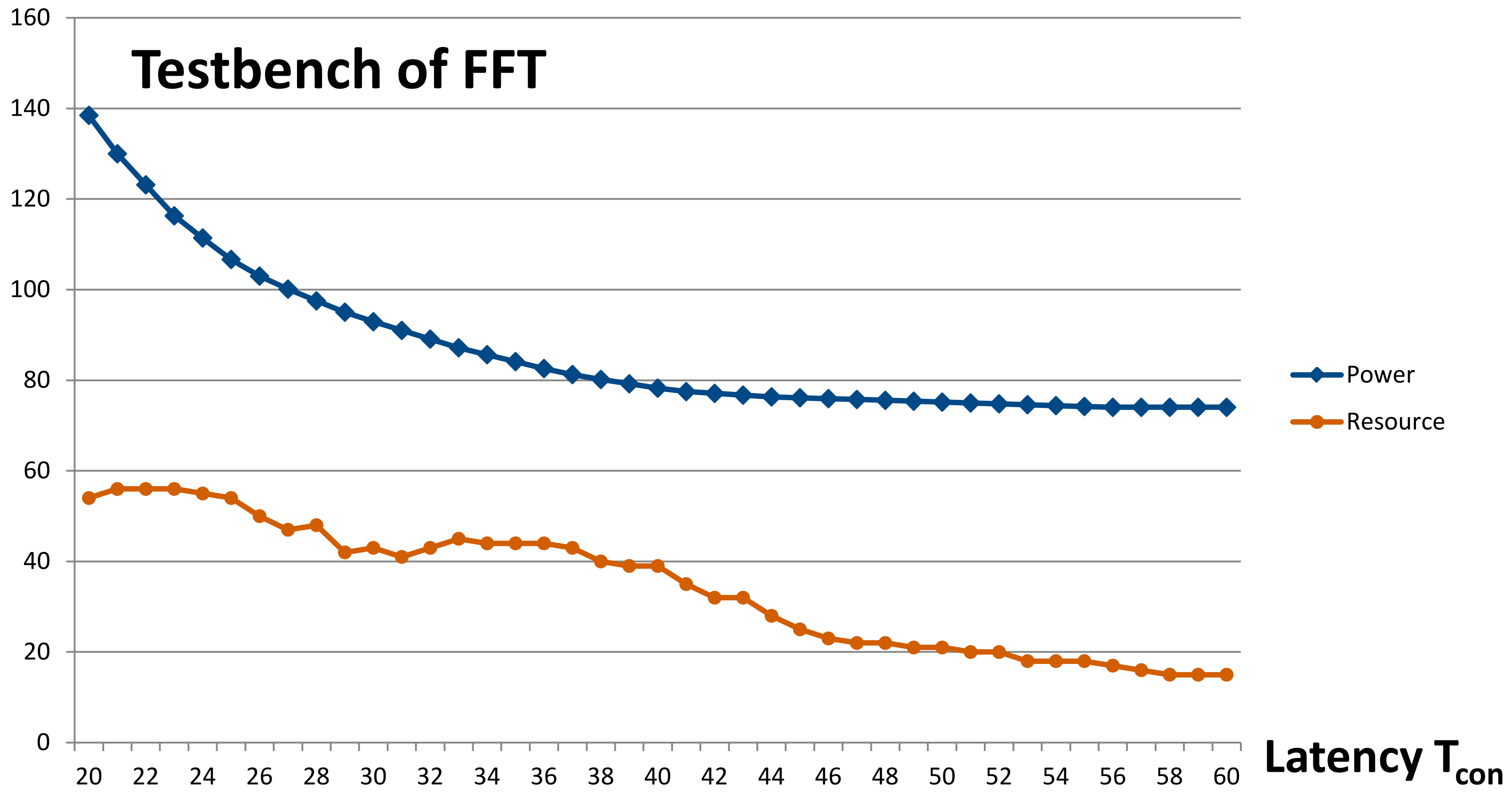
Optimality of Proposed NPMVS

- Longest running time of NPMVS is **0.25sec**(FFT under $T_{con}=2T$)
- Average running time of NPMVS is less than **0.1sec**, which is thousands times faster than ILP
- NPMVS get **optimum solutions** for **all the test benches**

DFG	T_{con}	NPMVS power(μw)	NPMVS resource	4d*	3d*	2d*	3d+	2d+	1d+	ILP power(μw)	ILP resource	4d*	3d*	2d*	3d+	2d+	1d+	Cmp power	Cmp resource
ad	T	66223	21	1	2	3	0	1	2	66223	21	1	2	3	0	1	2	1.0	1.0
	1.5T	36771	20	2	3	1	2	0	0	36771	20	2	3	1	2	0	0	1.0	1.0
	2T	29069	15	3	1	0	2	1	0	29069	15	3	1	0	2	1	0	1.0	1.0
ae	T	72682	19	2	1	2	1	1	2	72682	19	2	1	2	1	1	2	1.0	1.0
	1.5T	41020	15	1	2	1	1	2	0	NA	NA	-	-	-	-	-	-	-	-
	2T	33040	15	3	1	0	2	1	0	NA	NA	-	-	-	-	-	-	-	-
ar	T	41056	27	4	0	4	1	0	2	41056	27	4	0	4	1	0	2	1.0	1.0
	1.5T	26600	23	2	4	0	1	2	2	26600	23	2	4	0	1	2	2	1.0	1.0
	2T	18572	15	4	0	0	1	2	0	18572	15	4	0	0	1	2	0	1.0	1.0
diff	T	10955	10	0	2	1	0	0	1	10955	10	0	2	1	0	0	1	1.0	1.0
	1.5T	7523	10	2	0	1	0	1	0	7523	10	2	0	1	0	1	0	1.0	1.0
	2T	5359	7	2	0	0	0	1	0	5359	7	2	0	0	0	1	0	1.0	1.0
ellip	T	49872	19	1	1	3	0	2	2	49872	19	1	1	3	0	2	2	1.0	1.0
	1.5T	27220	18	2	2	1	1	2	0	27220	18	2	2	1	1	2	0	1.0	1.0
	2T	21190	17	4	1	0	2	0	0	21190	17	4	1	0	2	0	0	1.0	1.0
mpeg	T	77423	10	1	0	1	1	1	2	77423	10	1	0	1	1	1	2	1.0	1.0
	1.5T	52147	10	1	0	1	1	2	1	NA	NA	-	-	-	-	-	-	-	-
	2T	36931	10	1	1	0	2	2	0	NA	NA	-	-	-	-	-	-	-	-
fft	T	144615	50	4	6	2	0	4	10	NA	NA	-	-	-	-	-	-	-	-
	1.5T	102503	46	6	4	1	0	12	1	NA	NA	-	-	-	-	-	-	-	-
	2T	93413	20	4	0	0	8	0	0	NA	NA	-	-	-	-	-	-	-	-
AVR																		**1.0**	**1.0**

Latency vs Power & Resource

- Power, Resource are evaluated for data "FFT"
  - changing the latency constraint T_{con}
- Power decreases constantly, but resource doesn't



Conclusion

- **Network Simplex** Method
  - By analyzing the constraint matrix of MVS problem, “Piecewise-Linear extended Network Simplex Method (**PLNSM**)” is proposed on a **mobility graph**
- **Two Stage Heuristic** method
  - MVS problem is partitioned into “**Extended Power Minimization**” and “**Dependency-free Scheduling**” problems by introducing a variable “**Vertex Potential**”

Thank you!