Solution Methodologies for Integrated Network Design and Scheduling Problems

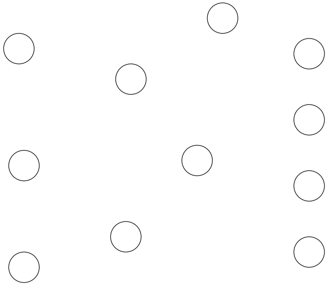
Sarah G. Nurre and Thomas C. Sharkey

Los Alamos National Laboratory
Student Smart Grid Seminar

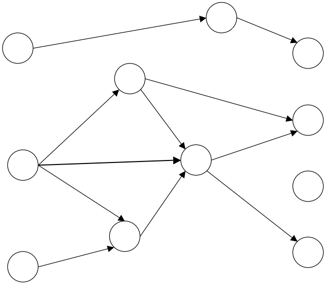
June 28, 2011

What are Integrated Network Design and Scheduling (INDS) Problems?

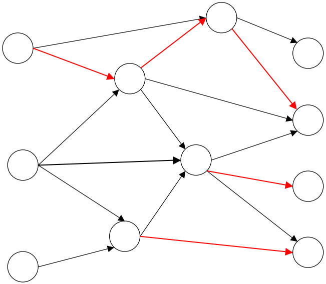
What are Integrated Network Design and Scheduling (INDS) Problems?



What are Integrated Network Design and Scheduling (INDS) Problems?

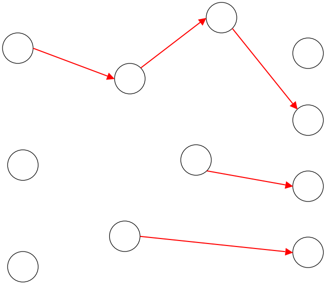


What are Integrated Network Design and Scheduling (INDS) Problems?

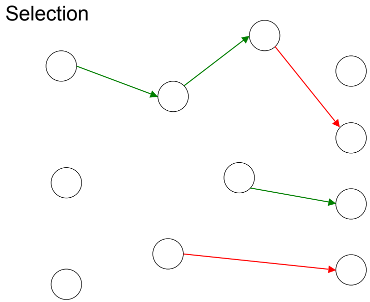


What are Integrated Network Design and Scheduling (INDS) Problems?

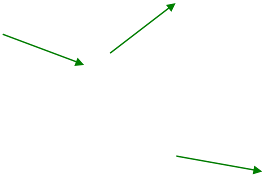
What are Integrated Network Design and Scheduling (INDS) Problems?



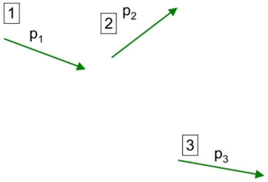
What are Integrated Network Design and Scheduling (INDS) Problems?



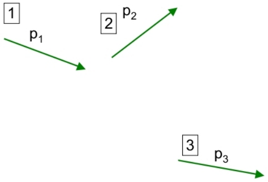
What are Integrated Network Design and Scheduling (INDS) Problems?



What are Integrated Network Design and Scheduling (INDS) Problems?

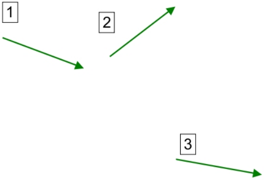


What are Integrated Network Design and Scheduling (INDS) Problems?



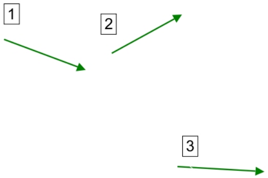
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



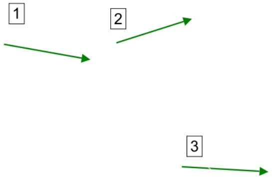
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



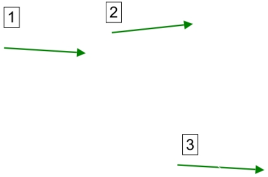
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



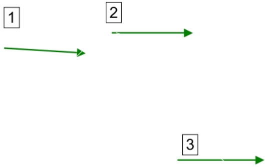
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



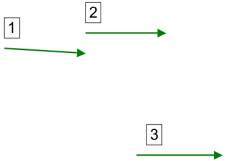
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



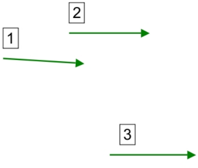
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



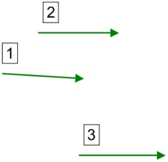
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



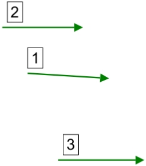
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



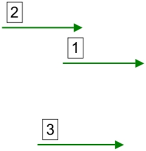
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?



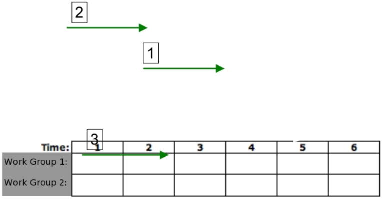
Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

What are Integrated Network Design and Scheduling (INDS) Problems?

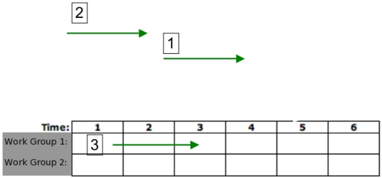


Time:	1	2	3	4	5	6
Work Group 1:						
Work Group 2:						

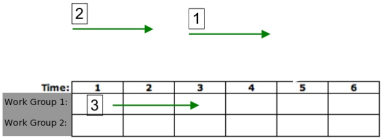
What are Integrated Network Design and Scheduling (INDS) Problems?



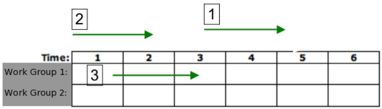
What are Integrated Network Design and Scheduling (INDS) Problems?



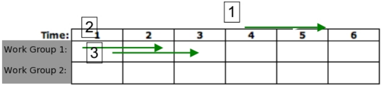
What are Integrated Network Design and Scheduling (INDS) Problems?



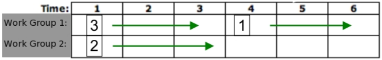
What are Integrated Network Design and Scheduling (INDS) Problems?



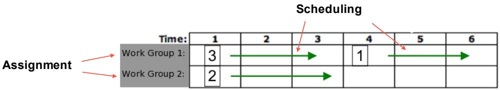
What are Integrated Network Design and Scheduling (INDS) Problems?



What are Integrated Network Design and Scheduling (INDS) Problems?



What are Integrated Network Design and Scheduling (INDS) Problems?



Decisions for an Integrated Network Design and Scheduling Problem

- Three main decisions occurring *simultaneously*:
  - Selection: select arcs to install into the network
  - Assignment: assign arcs to a work group
  - Scheduling: schedule arcs assigned to a work group

Motivation and Applications for Network-based Scheduling Problems

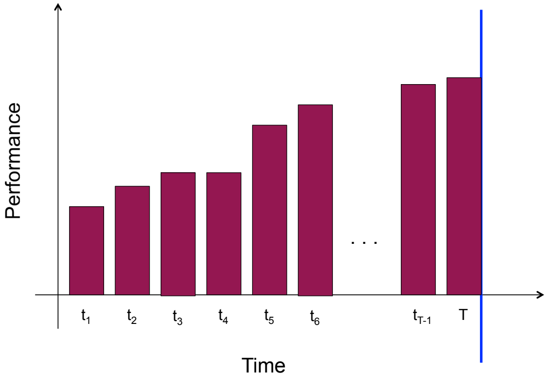
- Infrastructure Restoration after extreme events
  - Install components (either arcs or nodes) into the networks.
  - Power, telecommunications, water, waste water, and transportation networks.
- Humanitarian Logistics
  - Most often install nodes but also can install arcs into the network.
  - Location set-up for distribution of essential products such as food and water.
- PHEV Exchange Station Operations
  - Schedule the charging and discharging of batteries within an exchange station

Metrics for Network Performance Evaluation

- Maximum Flow
- Minimum Cost Flow
- Shortest Path
- Minimum Spanning Tree

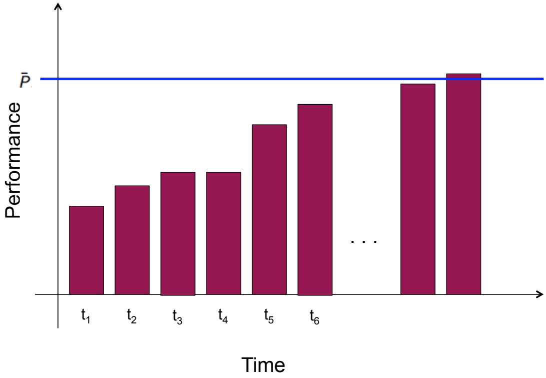
Objective Functions

- Cumulative objective
  - Set time horizon T.
  - Sum performance of the network for $t = 1, \ldots, T$, $\sum_{t=1}^{T} w_t P(G(t))$.



Objective Functions

- Makespan objective
  - Set performance value, $\bar{P}$.
  - Minimize the amount of time needed to reach or exceed $\bar{P}$.



Complexity Results Overview

Problem	Objective Function	NP-Hard	Strongly NP-Hard	Approximation ln(n)
Minimum Cost Flow	Cumulative		x	
	Makespan			x
Maximum Flow	Cumulative		x	
	Makespan			x
Shortest Path - Multiple Destinations	Cumulative		x	
	Makespan			x
Shortest Path - Multiple, Max Path	Cumulative		x	
	Makespan			x
Shortest Path - Single Destination	Cumulative			
	Makespan	x		
Minimum Spanning Tree	Cumulative			
	Makespan	x		

Weighted Shortest Processing Time (WSPT) Dispatching Rule

- Have a set of jobs J that need to be completed on M work groups.
- Each job $j \in J$ has an associated weight w_j that represents the added value to the objective or priority level.
- Each job $j \in J$ has an associated processing time p_j.
- How do we decide which job should be scheduled next?

Weighted Shortest Processing Time (WSPT) Dispatching Rule

- Have a set of jobs J that need to be completed on M work groups.
- Each job $j \in J$ has an associated weight w_j that represents the added value to the objective or priority level.
- Each job $j \in J$ has an associated processing time p_j.
- How do we decide which job should be scheduled next?
- WSPT Rule selects the job to be processed next that maximizes:

$$\frac{w_j}{p_j} \text{ for } j \in J$$

Adapt the WSPT Rule to solve INDS Problems

- What are the jobs?
- What are the weights of jobs, w_j?
- What the processing times of jobs, p_j?
- How do you find the job that maximizes $\frac{w_j}{p_j}$?
- We need to make sure to answer the 3 main decisions: selection, assignment, and scheduling.

Maximum Flow NBSP Dispatching Rule

- Optimality Conditions: Flow is maximum if and only if there does not exist an augmenting path from source to sink with a residual capacity greater than 0.

Maximum Flow NBSP Dispatching Rule

- Optimality Conditions: Flow is maximum if and only if there does not exist an augmenting path from source to sink with a residual capacity greater than 0.
- What are the jobs?
  - View augmenting paths as jobs.

Maximum Flow NBSP Dispatching Rule

- Optimality Conditions: Flow is maximum if and only if there does not exist an augmenting path from source to sink with a residual capacity greater than 0.
- What are the jobs?
  - View augmenting paths as jobs.
- What are the weights of jobs, w_j?
  - Flow on the path = minimum residual capacity of arcs in path.

Maximum Flow NBSP Dispatching Rule

- Optimality Conditions: Flow is maximum if and only if there does not exist an augmenting path from source to sink with a residual capacity greater than 0.
- What are the jobs?
  - View augmenting paths as jobs.
- What are the weights of jobs, w_j?
  - Flow on the path = minimum residual capacity of arcs in path.
- What the processing times of jobs, p_j?
  - Sum of the processing times of arcs in the path.

Maximum Flow NBSP Dispatching Rule

- How do you find the job that maximizes $\frac{w_j}{p_j}$?

Maximum Flow NBSP Dispatching Rule

- How do you find the job that maximizes $\frac{w_j}{p_j}$?
- Observation:
  - If we know the optimal numerator, w_j (min residual capacity).
  - Alter network to only look at arcs with residual capacity greater than or equal to this value.
  - Shortest path in this altered network maximizes $\frac{w_j}{p_j}$.

Maximum Flow NBSP Dispatching Rule

- How do you find the job that maximizes $\frac{w_j}{p_j}$?
- Observation:
  - If we know the optimal numerator, w_j (min residual capacity).
  - Alter network to only look at arcs with residual capacity greater than or equal to this value.
  - Shortest path in this altered network maximizes $\frac{w_j}{p_j}$.
- Solve for shortest path for every possible numerator value.
- $O(m)$ shortest path problems to find job that maximizes $\frac{w_j}{p_j}$.

Maximum Flow NBSP Dispatching Rule

- How do you find the job that maximizes $\frac{w_j}{p_j}$?
- Observation:
  - If we know the optimal numerator, w_j (min residual capacity).
  - Alter network to only look at arcs with residual capacity greater than or equal to this value.
  - Shortest path in this altered network maximizes $\frac{w_j}{p_j}$.
- Solve for shortest path for every possible numerator value.
- $O(m)$ shortest path problems to find job that maximizes $\frac{w_j}{p_j}$.
- Three decision answers:
  - Selection: According to this algorithm, select a path
  - Assignment and Scheduling: Order arcs in path according to longest processing time first, and assign to the first machine or work group that becomes available.

Minimum Cost Flow INDS Dispatching Rule

- Optimality Conditions: Flow is the minimum cost flow if and only if there does not exist an a negative cycle in the residual network composed of arcs whose residual capacity is greater than 0.

Minimum Cost Flow INDS Dispatching Rule

- Optimality Conditions: Flow is the minimum cost flow if and only if there does not exist an a negative cycle in the residual network composed of arcs whose residual capacity is greater than 0.
- What are the jobs?
  - View negative cycles as jobs.

Minimum Cost Flow INDS Dispatching Rule

- Optimality Conditions: Flow is the minimum cost flow if and only if there does not exist an a negative cycle in the residual network composed of arcs whose residual capacity is greater than 0.

- What are the jobs?
  - View negative cycles as jobs.

- What are the weights of jobs, w_j?
  - Improvement in cost by pushing flow over this cycle = (minimum residual capacity of arcs in cycle) x (cost of cycle)

Minimum Cost Flow INDS Dispatching Rule

- Optimality Conditions: Flow is the minimum cost flow if and only if there does not exist an a negative cycle in the residual network composed of arcs whose residual capacity is greater than 0.
- What are the jobs?
  - View negative cycles as jobs.
- What are the weights of jobs, w_j?
  - Improvement in cost by pushing flow over this cycle = (minimum residual capacity of arcs in cycle) x (cost of cycle)
- What the processing times of jobs, p_j?
  - Sum of the processing times of arcs in the cycle.

Minimum Cost Flow INDS Dispatching Rule

- How do you find the job that minimizes $\frac{w_j}{p_j} = \frac{\sum c_j}{\sum p_j} \min r_j$?

Minimum Cost Flow INDS Dispatching Rule

- How do you find the job that minimizes $\frac{w_j}{p_j} = \frac{\sum c_j}{\sum p_j} \min r_j$?
- Observation:
  - Suppose we know the residual capacity of the optimal cycle.
  - Alter network to only look at arcs with residual capacity greater than or equal to this value.
  - We want to find the cycle that minimizes the ratio of cost to processing time.
  - This is the min cost to time ratio problem and can be solved using min cost flow algorithms.

Minimum Cost Flow INDS Dispatching Rule

- How do you find the job that minimizes $\frac{w_j}{p_j} = \frac{\sum c_j}{\sum p_j} \min r_j$?
- Observation:
  - Suppose we know the residual capacity of the optimal cycle.
  - Alter network to only look at arcs with residual capacity greater than or equal to this value.
  - We want to find the cycle that minimizes the ratio of cost to processing time.
  - This is the min cost to time ratio problem and can be solved using min cost flow algorithms.
- Solve the min cost to time ratio problem for every possible residual capacity.
- $O(m)$ min cost to time ratio problems to find job that minimizes $\frac{w_j}{p_j}$.

Shortest Path INDS Dispatching Rule

- We can solve the shortest path problem by solving a minimum cost flow problem.
- Source node supply = number of destination nodes.
- Each destination node has a demand of 1.
- The flow on arc (i, j) signifies the number of destination nodes that use arc (i, j) in their shortest path.
- Only O(number of destination nodes) possible residual capacity values.
- Solve O(number of destination nodes) min cost to time ratio problems to find the cycle that minimizes $\frac{w_j}{p_j}$.

Minimum Spanning Tree INDS Dispatching Rule

- Optimality Conditions: A tree is the minimum spanning tree if and only if for every non-tree arc (i, j), its cost is greater than or equal to the cost of every arc on the unique path from i to j in the tree.

Minimum Spanning Tree INDS Dispatching Rule

- Optimality Conditions: A tree is the minimum spanning tree if and only if for every non-tree arc (i, j), its cost is greater than or equal to the cost of every arc on the unique path from i to j in the tree.
- What are the jobs?
  - An individual arc.

Minimum Spanning Tree INDS Dispatching Rule

- Optimality Conditions: A tree is the minimum spanning tree if and only if for every non-tree arc (i, j), its cost is greater than or equal to the cost of every arc on the unique path from i to j in the tree.

- What are the jobs?
  - An individual arc.
- What are the weights of jobs, w_j?
  - Max cost of arc in path from (i, j) in MST - $c_{(i,j)}$.

Minimum Spanning Tree INDS Dispatching Rule

- Optimality Conditions: A tree is the minimum spanning tree if and only if for every non-tree arc (i, j), its cost is greater than or equal to the cost of every arc on the unique path from i to j in the tree.
- What are the jobs?
  - An individual arc.
- What are the weights of jobs, w_j?
  - Max cost of arc in path from (i, j) in MST - $c_{(i,j)}$.
- What the processing times of jobs, p_j?
  - Processing time of the arc.

Minimum Spanning Tree INDS Dispatching Rule

- Optimality Conditions: A tree is the minimum spanning tree if and only if for every non-tree arc (i, j), its cost is greater than or equal to the cost of every arc on the unique path from i to j in the tree.
- What are the jobs?
  - An individual arc.
- What are the weights of jobs, w_j?
  - Max cost of arc in path from (i, j) in MST - $c_{(i,j)}$.
- What the processing times of jobs, p_j?
  - Processing time of the arc.
- How do you find the job that minimizes $\frac{w_j}{p_j}$?
  - $O(m)$ searches to find paths.

Case Study: Lower Manhattan Power Network

- Data:
  - Represents the realistic power network of Lower Manhattan
  - 1603 Nodes
  - 2621 Arcs
- Tests:
  - Randomly ‘knock out’ 25%, 50%, and 75% of the arcs.
  - Randomly assign a processing time of 1, 2, or 3 to ‘knocked out’ arcs.
  - Implemented the Maximum Flow cumulative and makespan dispatching rules.
  - Compared to the Integer Programming formulation using CPLEX 12.0.
  - Constant weights $w_1 = w_2 = \ldots = w_T = 1$.
  - Tested for 1, 2, and 3 work groups.
  - Tested for two different makespan performance goals $\bar{P} = 75\%$ of damage and $\bar{P} = 100\% =$ max flow.
  - Results presented are the average over 5 random instances.

Case Study: Cumulative Results

		Dispatching Rule		CPLEX 12.0	
Work Groups	Percentage	Time (s)	Performance (%)	Time (s)	Performance(%)
1	25%	28.75	0.72%	12335	0.34%
	50%	31.97	0.07%	5178	0.03%
	75%	19.33	0.40%	14400	0.40%
2	25%	32.13	0.85%	14400	0.66%
	50%	68.10	0.15%	7363	0.03%
	75%	43.22	0.39%	12100	0.27%
3	25%	29.33	0.61%	14400	0.51%
	50%	139.59	0.67%	14400	0.46%
	75%	57.45	0.51%	14400	0.35%

Case Study: Makespan 75% Results

		Dispatching Rule		CPLEX 12.0	
Work Groups	Percentage	Time (s)	Performance (%)	Time (s)	Performance(%)
1	25%	18.90	2.34%	1.37	0.00%
	50%	61.35	1.32%	3.79	0.00%
	75%	87.02	0.36%	8.29	0.00%
2	25%	16.63	3.27%	2881.07	0.91%
	50%	57.87	1.82%	2888.42	0.27%
	75%	81.84	0.70%	3916.89	0.22%
3	25%	18.77	6.36%	2882.46	1.33%
	50%	82.56	1.56%	98.60	0.00%
	75%	98.73	0.88%	11521.32	0.71%

Case Study: Makespan 100% Results

		Dispatching Rule		CPLEX 12.0	
Work Groups	Percentage	Time (s)	Performance (%)	Time (s)	Performance(%)
1	25%	21.03	4.88%	1.61	0.00%
	50%	112.35	6.99%	1347.62	0.00%
	75%	130.07	7.11%	13759.31*	1.33%
2	25%	18.93	6.70%	6155.98	0.98%
	50%	83.55	10.81%	14400	3.44%
	75%	136.73	11.05%	14400	3.17%
3	25%	16.68	10.01%	8677.14	2.34%
	50%	75.89	11.46%	14400	3.50%
	75%	121.09	12.04%	14400	3.15%

* For two instances CPLEX ran into a memory error before the 4 hour time limit.

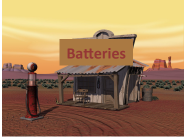
Conclusions

- Looked at the new class of problems integrated Network Design and Scheduling problems
- Created Novel Dispatching Rules based on Optimality Conditions of the problems
- Applied Dispatching Rule to real life data set and provided near-optimal solutions quickly

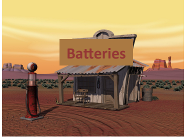
Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

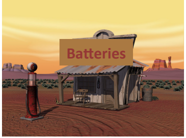


Basic PHEV Exchange Station Model



⋮

Basic PHEV Exchange Station Model



⋮

Basic PHEV Exchange Station Model

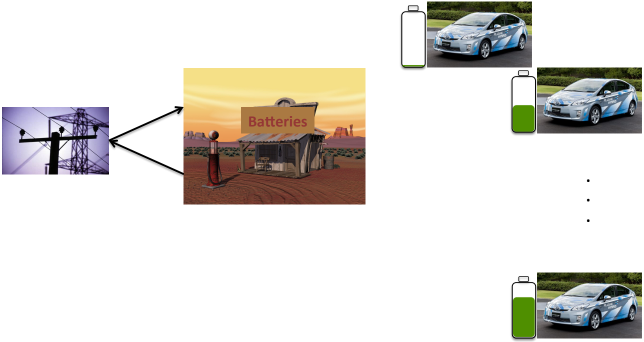
Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

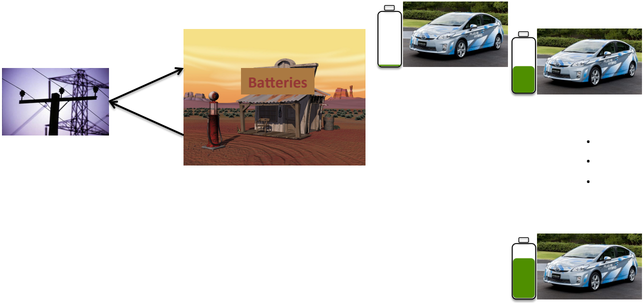
Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model



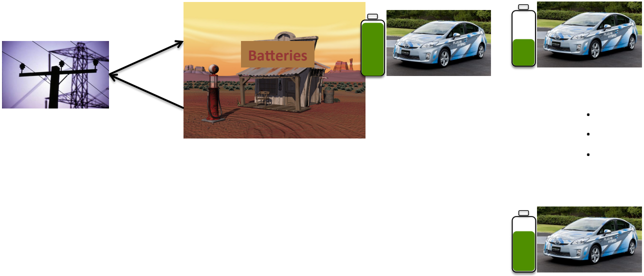
Basic PHEV Exchange Station Model



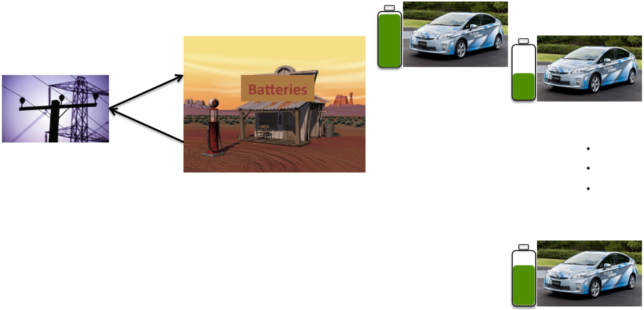
Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model



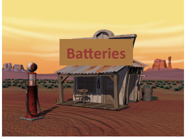
Basic PHEV Exchange Station Model



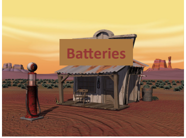
Basic PHEV Exchange Station Model

Basic PHEV Exchange Station Model

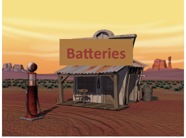
Basic PHEV Exchange Station Model



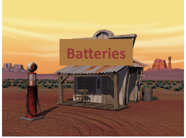
Basic PHEV Exchange Station Model

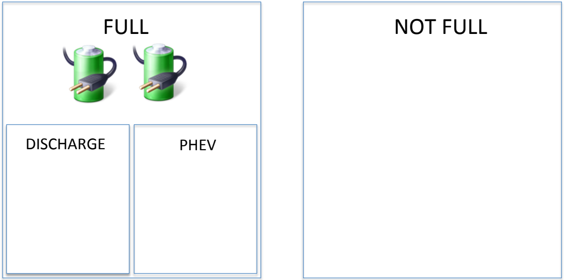


Basic PHEV Exchange Station Model

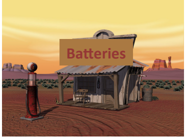


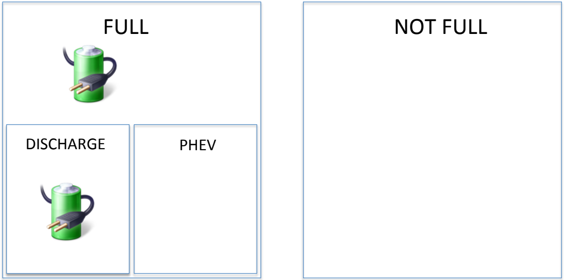
Basic PHEV Exchange Station Model



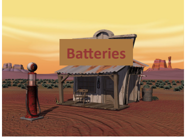


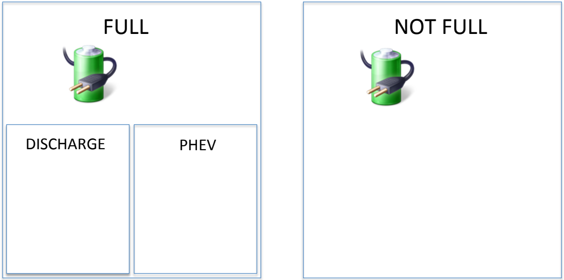
Basic PHEV Exchange Station Model



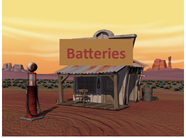


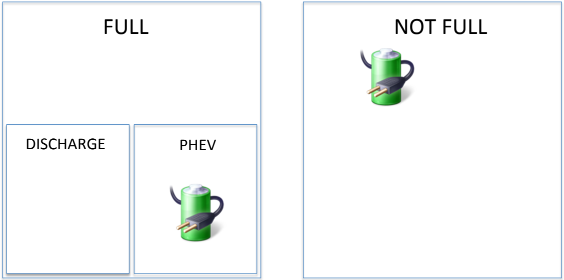
Basic PHEV Exchange Station Model



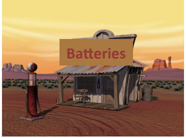


Basic PHEV Exchange Station Model





Basic PHEV Exchange Station Model



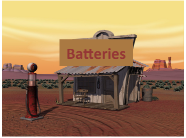
FULL

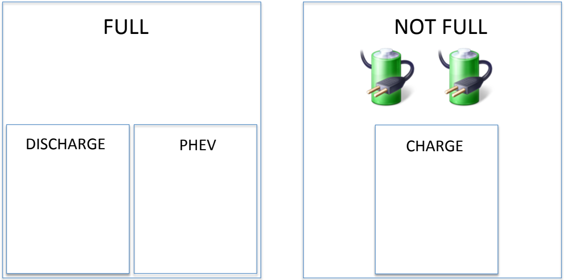
DISCHARGE

PHEV

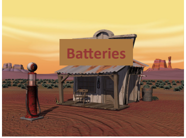
NOT FULL

Basic PHEV Exchange Station Model





Basic PHEV Exchange Station Model



FULL

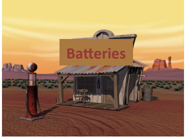
DISCHARGE

PHEV

NOT FULL

CHARGE

Basic PHEV Exchange Station Model



FULL

DISCHARGE

PHEV

NOT FULL

CHARGE

Basic PHEV Exchange Station Model

- Model the Exchange Station operations
- Determine the optimal number of batteries to charge, discharge, and exchange at each time
- Stochastic model with variable battery demand and power network flow
- Consider three different models that factor in different objective functions
  - One solution factoring in all different scenarios for the stochastic input
  - A solution for each scenario
  - A solution for each node of the tree that incorporates all scenarios

Reduction of Set Cover Problem to Maximum Flow INDS

- Show a reduction of known Strongly NP-Complete problem
- Set Cover Problem
  - Set of elements $X = \{x_1, x_2, \ldots, x_n\}$.
  - Set of sets $S = \{S_1, S_2, \ldots, S_m\}$.
  - Each $S_i \subseteq X$, for example $S_1 = \{x_1, x_2, x_4\}$.
  - Want to select the least amount of sets S_i such that each x_i is selected, i.e. select a set $\bar{S} \subseteq S$ such that $\cup_{S_i \in \bar{S}} S_i = X$.

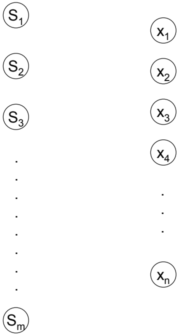
Reduction of Set Cover Problem to Maximum Flow INDS

- Show a reduction of known Strongly NP-Complete problem
- Set Cover Problem
  - Set of elements $X = \{x_1, x_2, \ldots, x_n\}$.
  - Set of sets $S = \{S_1, S_2, \ldots, S_m\}$.
  - Each $S_i \subseteq X$, for example $S_1 = \{x_1, x_2, x_4\}$.
  - Want to select the least amount of sets S_i such that each x_i is selected, i.e. select a set $\bar{S} \subseteq S$ such that $\cup_{S_i \in \bar{S}} S_i = X$.
- Use this reduction idea for:
  - Max Flow Cumulative and Makespan
  - Min Cost Flow Cumulative and Makespan
  - Shortest Path Multiple Destinations Cumulative and Makespan
  - Shortest Path Multiple, Max Path Cumulative and Makespan

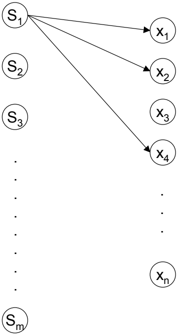
Formulation of Set Cover as a Max Flow INDS

Formulation of Set Cover as a Max Flow INDS

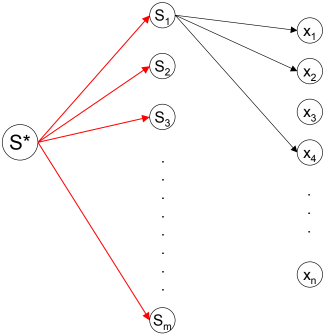
Formulation of Set Cover as a Max Flow INDS



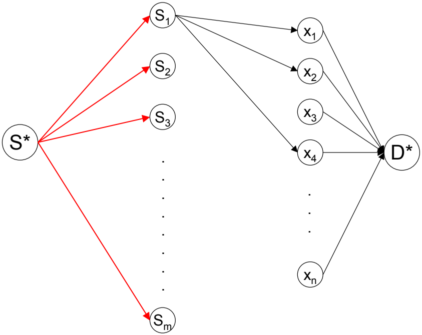
Formulation of Set Cover as a Max Flow INDS



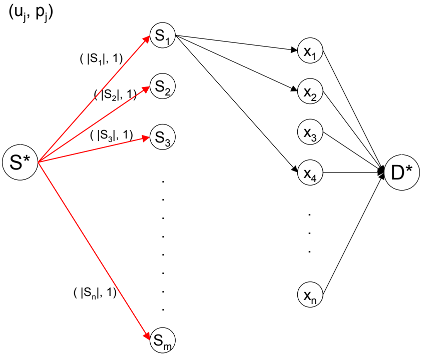
Formulation of Set Cover as a Max Flow INDS



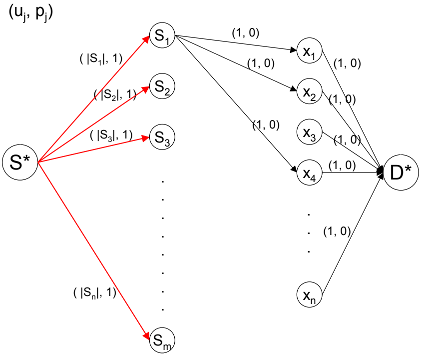
Formulation of Set Cover as a Max Flow INDS



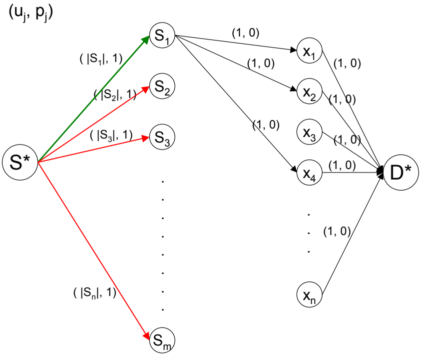
Formulation of Set Cover as a Max Flow INDS



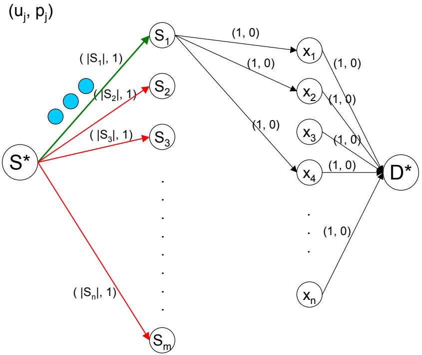
Formulation of Set Cover as a Max Flow INDS



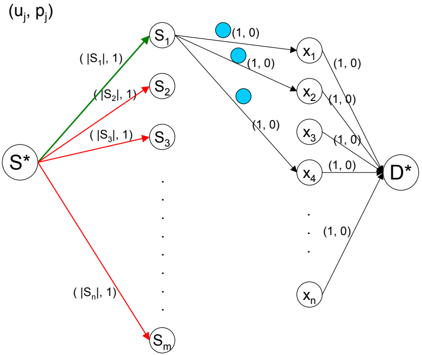
Formulation of Set Cover as a Max Flow INDS



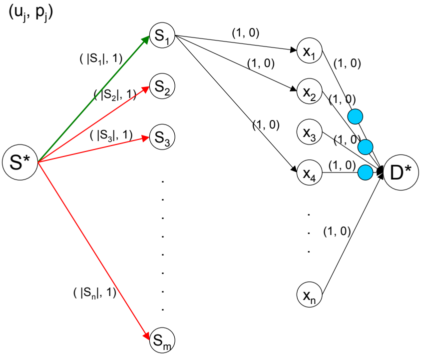
Formulation of Set Cover as a Max Flow INDS



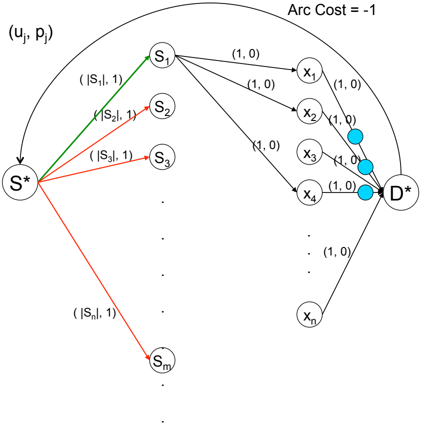
Formulation of Set Cover as a Max Flow INDS



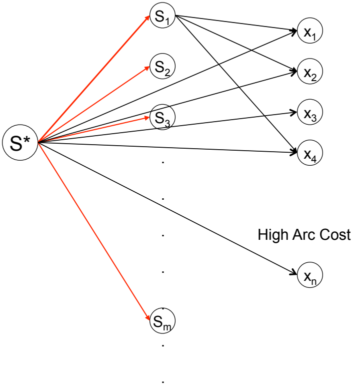
Formulation of Set Cover as a Max Flow INDS



Formulation of Set Cover as a Min Cost Flow INDS



Formulation of Set Cover as a Shortest Path INDS



Theorem

The Integrated Network Design and Scheduling Problem (INDS) with maximum flow cumulative performance objective is Strongly NP-Complete with one work group.

Theorem

The Integrated Network Design and Scheduling Problem (INDS) with maximum flow cumulative performance objective is Strongly NP-Complete with one work group.

Proof: Reduction of Set Cover to Maximum Flow INDS

- Solution to this formulation of INDS $\rightarrow$ Decision for Set Cover problem
- Set $w_1 = w_2 = \ldots = w_{T-1} = 0$, and $w_T = 1$
- Cumulative objective = max flow at time T
- If the max flow at time T equals $|X| \rightarrow$ A set cover is possible with $\leq T$ sets
- Conclusion: Maximum Flow Cumulative INDS is Strongly NP-Complete

Theorem

The Integrated Network Design and Scheduling Problem (INDS) with maximum flow makespan performance objective is Strongly NP-Complete with one work group.

Theorem

The Integrated Network Design and Scheduling Problem (INDS) with maximum flow makespan performance objective is Strongly NP-Complete with one work group.

Proof:

- Use same reduction
- Set performance value $\bar{P} = |X|$
- Minimum makespan value T' for INDS $\rightarrow$ minimum number of sets needed to cover all elements in X
- Conclusion: Maximum Flow Makespan INDS is Strongly NP-Complete

Theorem

The INDS with minimum cost flow cumulative and makespan performance objectives is Strongly NP-Complete with one work group.

Proof: Can formulate a Maximum Flow problem as a Minimum Cost Flow problem

Theorem

The INDS with multiple destination shortest path cumulative and makespan performance objectives is Strongly NP-Complete with one work group.

Proof:

- If cumulative shortest path equals n, a set cover is possible in T time.
- Makespan objective equals the minimum number of sets needed to cover all elements.

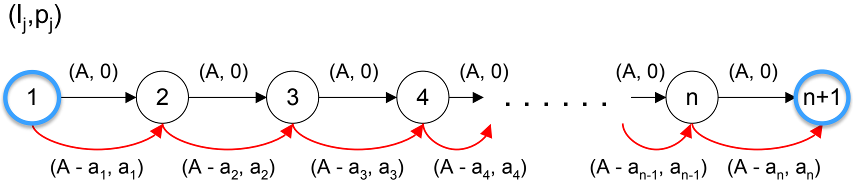
Reduce Partition Problem to Shortest Path Single Destination Makespan INDS

- Show a reduction of known NP-Complete problem to INDS.
- Partition Problem
  - Set of elements $S = \{a_1, a_2, \ldots, a_n\}$.
  - Is there $\bar{S} \subset S$ such that $\sum_{a_i \in \bar{S}} a_i = \sum_{a_j \notin \bar{S}} a_j = \sum_{k=1}^{n} \frac{a_k}{2}$.

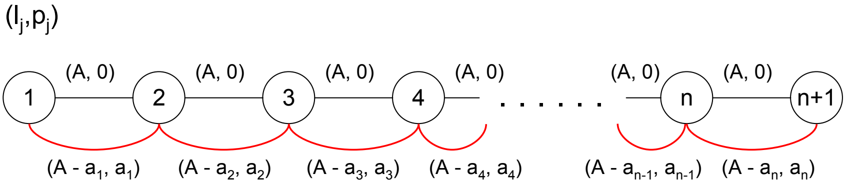
Reduce Partition Problem to Shortest Path Single Destination Makespan INDS

- Show a reduction of known NP-Complete problem to INDS.
- Partition Problem
  - Set of elements $S = \{a_1, a_2, \ldots, a_n\}$.
  - Is there $\bar{S} \subset S$ such that $\sum_{a_i \in \bar{S}} a_i = \sum_{a_j \notin \bar{S}} a_j = \sum_{k=1}^{n} \frac{a_k}{2}$.
- Use this reduction idea for:
  - Shortest Path Single Destination Makespan objective
  - Minimum Spanning Tree Makespan objective

Formulation of Partition as a Shortest Path INDS



Formulation of Partition as a MST INDS



Theorem

The INDS with shortest path single destination makespan performance objective is NP-Complete with one work group.

Proof: Reduction of Partition to Shortest Path INDS

- Set Makespan performance value to be $nA - \sum_{i=1^n} \frac{a_i}{2}$.
- If makespan objective equals $\sum_{i=1^n} \frac{a_i}{2}$ then we know a partition is possible.

Theorem

The INDS with minimum spanning tree makespan performance objective is NP-Complete with one work group.

Proof: Use same idea with reduction from partition