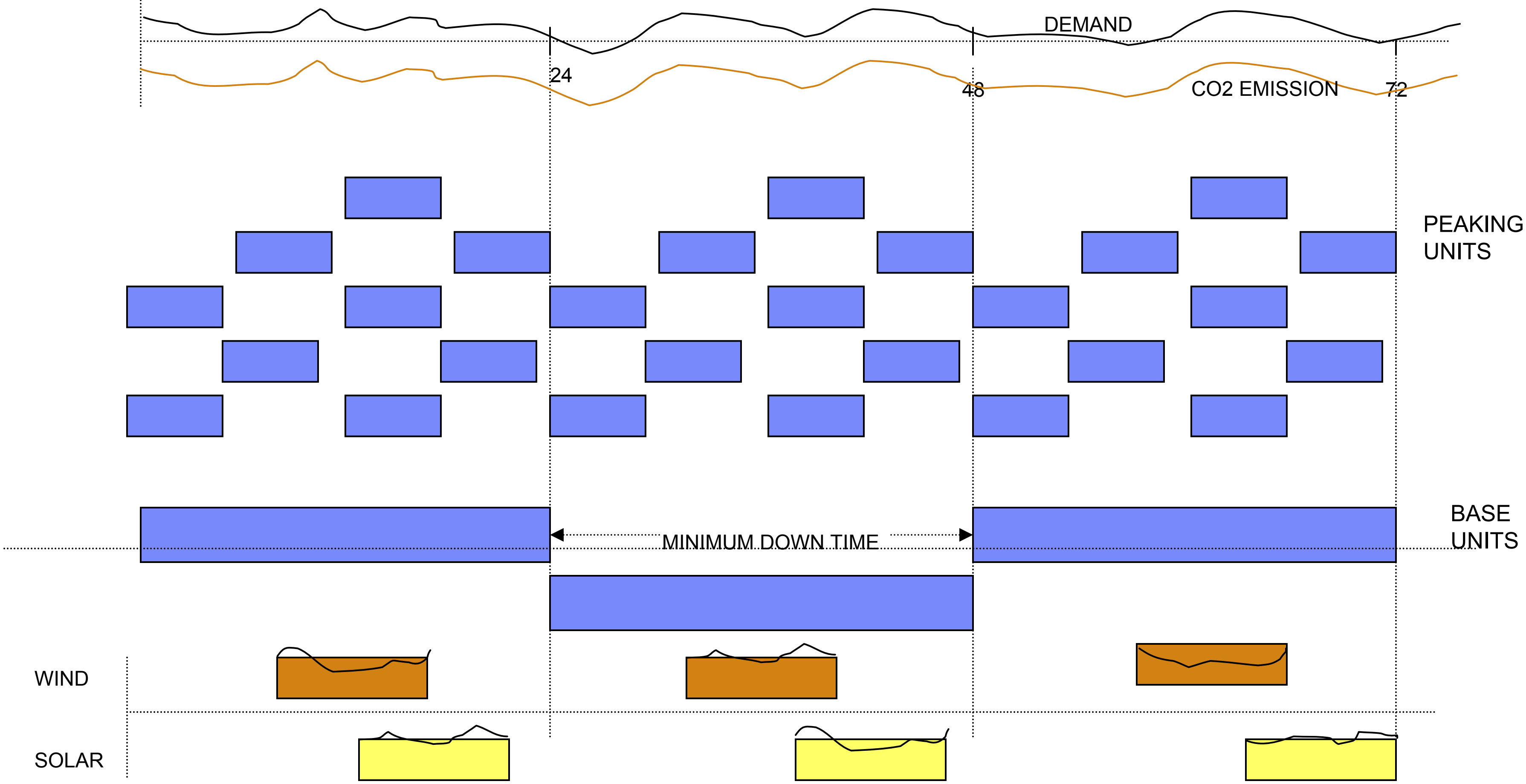
IBM Research

Stochastic Unit Commitment

Sanjeeb Dash, Joao Goncalves,
Jay(ant) Kalagnanam, Ali Koc, Ming Zhao,

BAMS, IBM Research
Contact: jayant@us.ibm.com

Unit Commitment Problem (Distributed Generation)



Integer programming problem with uncertain demand & supply
-> Stochastic optimization
The heat rate of a unit is a (nonlinear) function of load -> nonlinear optimization
- maintenance improves heat rate and hence CO2 emissions

Summary

- Formulations
- Primal heuristics for stochastic unit commitment
- Branch-and-cut for (stochastic) unit commitment
  - Cuts for linear-cost unit commitment
  - Cuts for nonlinear-cost unit commitment
  - Computational results
- Scenario generation using DeepThunder forecasts
- Stochastic unit commitment vs. spinning reserves
- Further research

Formulation:

$$
\begin{aligned}
\min \quad & \textstyle\sum_{t=1}^{T} \sum_{i=1}^{n} (u_{it}F_i + s_{it}S_i + f_i(g_{it})) & \\
s.t. \quad & \textstyle\sum_{i=1}^{n} g_{it} \geq d_t & \forall t \\
& q_i u_{it} \leq g_{it} \leq Q_i u_{it} & \forall i, t \\
& \text{Minimal up/down constraints} & \\
& u_{it} - u_{i,t-1} \leq s_{it} & \forall i, t \\
& u_{it}, s_{it} \in \{0, 1\} & \forall i, t.
\end{aligned}
$$

Variables:

- g_{it} is generation provided by unit i in period t,
- u_{it} is a binary variable indicating if unit i is up in period t,
- s_{it} is a binary variable indicating if unit i is started in period t.

Data:

- F_i is no-load cost of unit i's offer,
- S_i is startup cost of unit i's offer,
- $f_i(\cdot)$ is cost function of unit i,
- Q_i and q_i are the maximum/minimum generating capacity of unit i's offer,
- d_t is the load forecast in period t.

Minimum up/down Constraints

Formulation:

$$u_{i,t} - u_{i,t-1} \le u_{i,\tau} \quad \tau \in \{t+1, \ldots, \min\{t+L_i, T\}\} \ \forall i, t$$
$$u_{i,t-1} - u_{i,t} \le 1 - u_{i,\tau} \quad \tau \in \{t+1, \ldots, \min\{t+l_i, T\}\} \ \forall i, t.$$

where

L_i and l_i are the minimum up/down time for generator i when it is started up or shut down respectively.

A better formulation (D. Rajan and S. Takriti (2005))

$$\sum_{\tau=t-L_i+1}^{t} s_{i,\tau} \le u_{i,t} \quad \forall i, t, s$$
$$\sum_{\tau=t-l_i+1}^{t} s_{i,\tau} \le 1 - u_{i,t-l_i} \quad \forall i, t, s.$$

This formulation improves computation time dramatically

Comparison of Improved Formulation

J. Goez et al (2008)

Instance	Formulation	LP Time	Total Time	B&B nodes
1	General	15.3	436	150
	Convex hull	0.13	1	0
2	General	14.7	314	10
	Convex hull	0.12	1	0
3	General	11.71	2240	510
	Convex hull	0.11	62	400
4	General	1.98	3290	590
	Convex hull	0.04	46	300

Some additional constraint classes

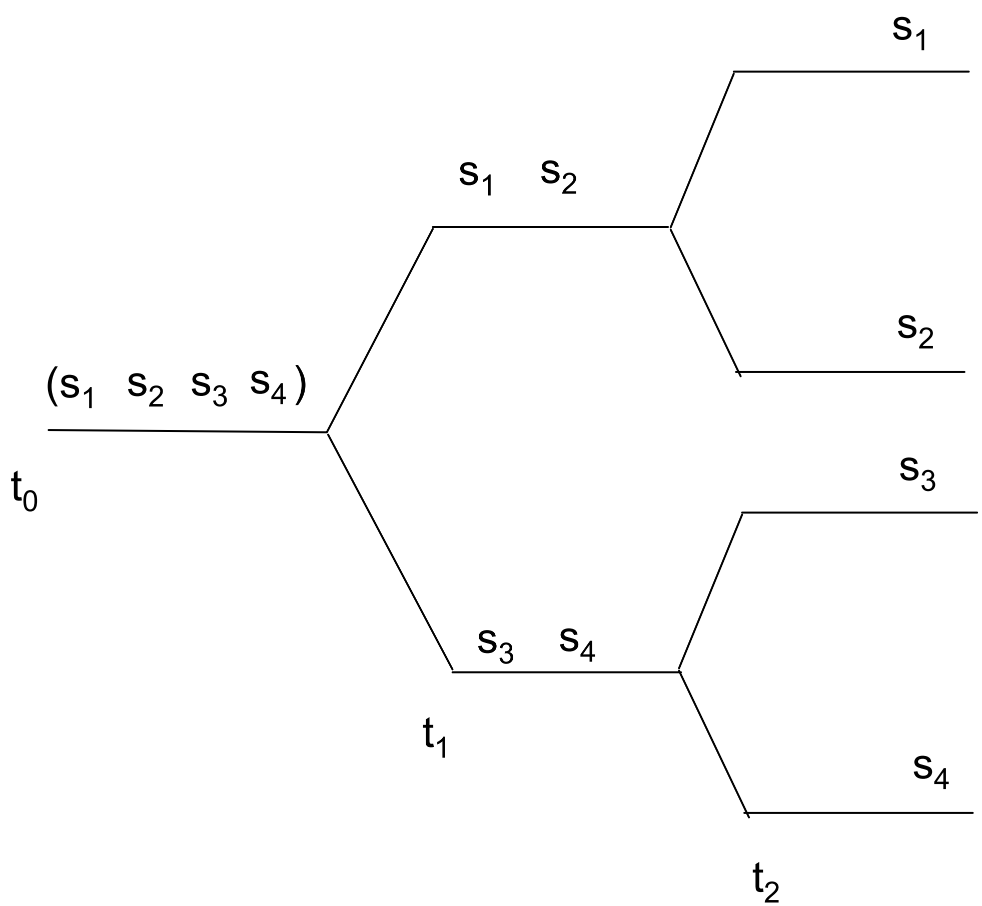
- **Ramping constraints**
  - $g_{i,t} + k_i \geq g_{i,t+1}$ $g_{i,t} - m_i \leq g_{i,t+1}$; k_i & m_i are ramp up/down rates
- **Spinning reserves**
- **Modeling storage**
- **Power flow constraints**
  - DC : Linear
  - AC: Nonlinear

Stochastic Unit Commitment with Linear Cost Functions

Formulation:

$$
\begin{array}{lll}
\min & \sum_s P_s \left(\sum_{t=1}^{T} \sum_{i=1}^{n} \left(u_{it} F_i + s_{it}^s S_i + c_i g_{it} \right) \right) & \\
s.t. & \sum_{i=1}^{n} g_{it}^s \geq d_t^s & \forall t, s \\
 & q_i u_{it}^s \leq g_{it} \leq Q_i u_{it}^s & \forall i, t, s \\
 & \text{Minimal up/down constraints} & \\
 & \text{Nonanticipativity constraints} & \\
 & u_{it}^s - u_{i,t-1}^s \leq s_{it}^s & \forall i, t, s \\
 & u_{it}^s \in \{0,1\}, s_{it}^s \in [0,1] & \forall i, t, s.
\end{array}
$$

Bundle (nonanticipativity) Constraints



If two scenarios are indistinguishable up to time *t*, then the decisions for both scenarios by time *t* should be the same.

If $d_t^{s_1} = d_t^{s_2} \; \forall t \leq \tau$, then we add equalities $u_{i,\tau}^{s_1} = u_{i,\tau}^{s_2} \; \forall i$. We call the collection of this equalities as nonanticipativity constraints

Unit Commitment with Nonlinear Cost Functions

Formulation:

$$
\begin{array}{lll}
\min & \sum_{t=1}^{T}\sum_{i=1}^{n}\left(u_{it}F_i + s_{it}S_i + \sum_{k=0}^{K} f_i(a_i^k)\lambda_{it}^k\right) & \\
s.t. & \sum_{i=1}^{n}\sum_{k=0}^{K} a_i^k\lambda_{it}^k \geq d_t & \forall t \\
 & q_i u_{it} \leq g_{it} \leq Q_i u_{it} & \forall i,t \\
 & \text{Minimal up/down constraints} & \\
 & \sum_{k=0}^{K}\lambda_{it}^k = 1 & \forall i,t \\
 & u_{it} - u_{i,t-1} \leq s_{it} & \forall i,t \\
 & \lambda_{it}^0,\ldots,\lambda_{it}^K \text{ is SOS2} & \forall i,t \\
 & \lambda_{it}^0, u_{it} \in \{0,1\}, \lambda_{it}^1,\ldots,\lambda_{it}^K, s_{it} \in [0,1] & \forall i,t.
\end{array}
$$

where $g_{i,t} = \sum_{k=0}^{K} a_i^k\lambda_{it}^k.$

Some heuristics to generate initial (feasible) solutions

- **Lagrangian relaxations**
  - Relax the bundle constraints and add a penalty term for violations
  - Solve the *s* subproblems independently as a starting point
- **LP rolling horizon heuristic**
  - Keep only one bundle as binary and relax the remaining
  - Start with t_0 and roll forward fixing previous periods
  - Provides good initial feasible solutions

LP rolling horizon heuristic

Scenarios	Time (s)	gap
2	88	0.00%
5	137	0.00%
8	137	0.03%
11	122	0.06%
14	131	0.05%
17	131	0.10%
20	149	0.10%
23	140	0.12%
26	155	0.13%

Semi-continuous Knapsack

We study PS = conv(S), where

$$S = \{ x \in R^n : \Sigma a_j x_j \geqslant b, x_j \in \{0\} \cup [l_j , u_j] \}$$

Assumption: $a_j = 1$

Proposition:

PS is full dimensional

(More general semi-continues cuts see I. R. de Farias " semi-continues cuts for Mixed-Integer Programming")

Cover Inequality

Definition:

Let $C \subseteq N$. We say that C is a cover if

$$\Sigma_{j \in C} u_j < b$$

Proposition: Let C be a cover. Then the inequality

$$\sum_{j \in N-C} \frac{x_j}{l_j} \geq 1$$

is valid for *PS*.

Zhao and Kalagnanam (09) strengthen this cover inequality, develops cover inequalities for semi-SOS2 knapsack, and proposes heuristic separation algorithms.

Computational Results (Linear cost function)

- CPLEX 11.0 is used as an LP solver,
- The instance: 32 units and 72 periods,
- The instance are terminated after at most 7,200 CPU seconds.
- Cover and flow cover cuts are turned off for testing instances with user cuts.

Computational Results – basic comparison

	CPLEX DS		CPLEX B&C			With user + mir cuts			Reduc.%	
Prob.	Node	Time	Node	Time	Gap%	Node	Time	# cuts	Node	Time
2.1	2,209	606	7,427	2,202		1,249	193	310	83.2	91.3
2.2	999	402	3,150	779		901	165	176	71.4	21.2
2.3	684	238	1,825	555		417	83	173	77.2	85.0
3.1	3,270	1,264	19,873	*	0.07	4,473	863	370		
3.2	967	431	8,063	2,725		1,790	248	284	77.8	90.9
3.3	1,136	667	10,317	*	0.06	1,577	400	338		
4.1	1,213	1,116	8,150	*	0.13	3,302	1,351	406		
4.2	1,334	918	13,968	*	0.06	2,503	613	412		
4.3	2,158	1,698	12,701	*	0.14	8,568	1,653	507		
5.1	4530	5892	7616	*	0.17	14970	5673	642		
5.2	1533	1977	6524	*	0.10	4010	1396	494		
5.3	2521	3108	7301	*	0.14	13471	2596	650		

4 scenarios, 2nd instance

- # All variables 14,000 – 35,000
- # Binary variables 4,600 – 11,500
- # Constraints 20,000 – 50,000

Computational Results – optimal solution

	CPLEX DS				CPLEX B&C				With user cuts		
Prob.	Node	Time	Gap%	Red.%	Node	Time	Gap%	Red.%	Node	Time	Gap%
2.1	2,209	606	0.01	79.9	4,137	1,223	0.09	90.0	640	122	0.07
2.2	979	395	0.00	77.5	1,000	296	0.12	70.0	440	89	0.09
2.3	320	126	0.21	54.8	1,000	340	0.08	83.2	250	57	0.04
3.1	2,934	1,153	0.01	25.6					4,410	858	0.07
3.2	967	431	0.00	83.1	2,189	877	0.12	91.7	310	73	0.09
3.3	905	557	0.21	47.0	4,100	2,694	0.08	89.0	839	295	0.04
4.1	640	623	0.06	33.1					605	417	0.08
4.2	974	642	0.03	24.0	4,000	1,740	0.12	72.0	1,552	488	0.02
4.3	752	653	0.09	-55.7	1,000	615	0.25	-65.3	4,000	1017	0.04
5.1	4072	5210	0.02	85.0	1100	1023	0.25	23.6	933	782	0.09
5.2	1048	1279	0.04	3.7	1000	1165	0.19	-5.7	3000	1232	0.02
5.3	1230	1352	0.08	28.0	3100	2631	0.18	63.0	2100	973	0.05

Computational Results – root node

	Without user cuts		With user cuts	
Prob.	Time	Gap%	Time	Gap%
2.1	20	0.77	19	0.41
2.2	29	0.34	17	0.28
2.3	27	0.83	19	0.13
3.1	77	0.41	37	0.34
3.2	42	0.69	18	0.12
3.3	41	0.97	25	0.20
4.1	80	0.43	37	2.42
4.2	53	0.70	30	0.21
4.3	82	0.43	48	0.28
5.1	88	0.49	75	0.36
5.2	97	2.71	74	0.20
5.3	97	2.80	89	0.21

Conclusions

- With user cuts, all the instance are solved to optimality and the time reduction is more than 80% in average.
- The difficulty lies on closing optimality gap.
- By adding cutting planes in the initial formulation, one can take advantages of dynamic search.

Computational Results –Nonlinear cost functions

- CPLEX 11.0 is used as an LP solver,
- The instance: 72 periods, and 4 segments in piecewise linear functions,
- The instance are terminated after at most 3,600 CPU seconds.
- SOS2 concept is enforced by introducing binary variables to take advantages of CPLEX, since the model already has lots of binary variables.

Computational Results – basic comparison

	CPLEX Default			With user + mir cuts			Reduc.%	
Prob.	Node	Time	Gap%	Node	Time	# cuts	Node	Time
7.1	7,727	21		208	2	673	97.3	90.5
7.2	1,154,200	*	0.01	5,569	17	762		
7.3	1,138,300	*	0.02	460	3	575		
X8.1	1,160,900	2,416	0.10	13,463	97	1,702		
8.2	21,500	68		49	2	787	99.8	97.1
8.3	132,700	524		676	6	1,080	99.5	98.9
X9.1	1,040,300	2,067	0.09	76,425	523	2,194		
X9.2	1,078,600	3,006	0.02	415	6	1,283		
X9.3	1,014,100	1,741	0.11	35,104	218	1,900		
10.1	873,400	*	0.08	2,464	69	2,379		
10.2	838,800	*	0.08	5,684	107	2,368		
X10.3	908,100	3,039	0.11	118,073	804	2,507		

10 units, 2nd instance

- # All variables 4,000 – 5,000
- # Binary variables 1,500 – 2,100
- # Constraints 4,200 – 6,000

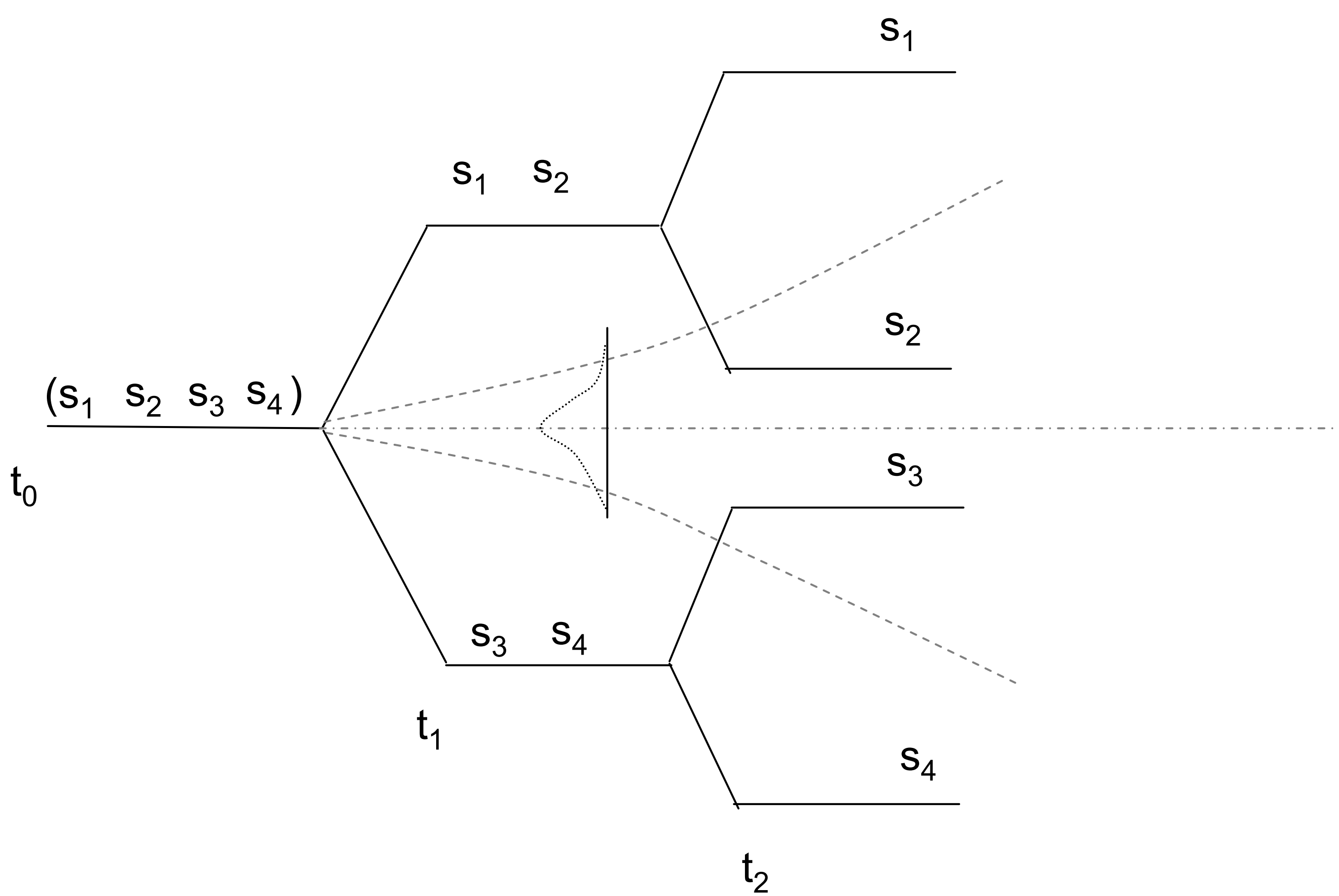
Computational Results – optimal solution

	CPLEX Default			With user cuts			Reduc.%
Prob.	Node	Time	gap%	Node	Time	gap%	Time
7.1	1,522	8	0.02	184	2	0.00	75.0
7.2	69,902	179	0.04	4,329	14	0.02	92.2
7.3	88,608	248	0.05	320	2	0.07	99.2
8.1	28,621	73	0.14	13,168	96	0.00	-31.5
8.2	583	6	0.04	48	1	0.00	83.3
8.3	36,185	117	0.02	407	4	0.03	96.6
9.1				71,940	488	0.01	
9.2				223	4	0.02	
9.3	1,009,196	3,246	0.11	30,896	193	0.01	94.1
10.1	5,524	38	0.23	2,374	67	0.01	-76.3
10.2	133,580	598	0.11	4,380	93	0.02	84.4
10.3				52,500	371	0.02	

Modeling Wind Intermittency

- **Demand is uncertain, i.e. d_t is a random variable**
- **Wind energy is forecast using weather models**
  - Wind speed and direction can be forecast but with uncertainty
  - For each farm, generation $g_{i,t}$ is a random variable
- **Assume that wind energy (subject to technical cut-in constraints) has to be used (regulatory)**
  - A must-take constraint
- **Therefore the total demand can be written as**
  - $D_t = d_t - \Sigma_i \, g_{i,t}$ (a new random variable)

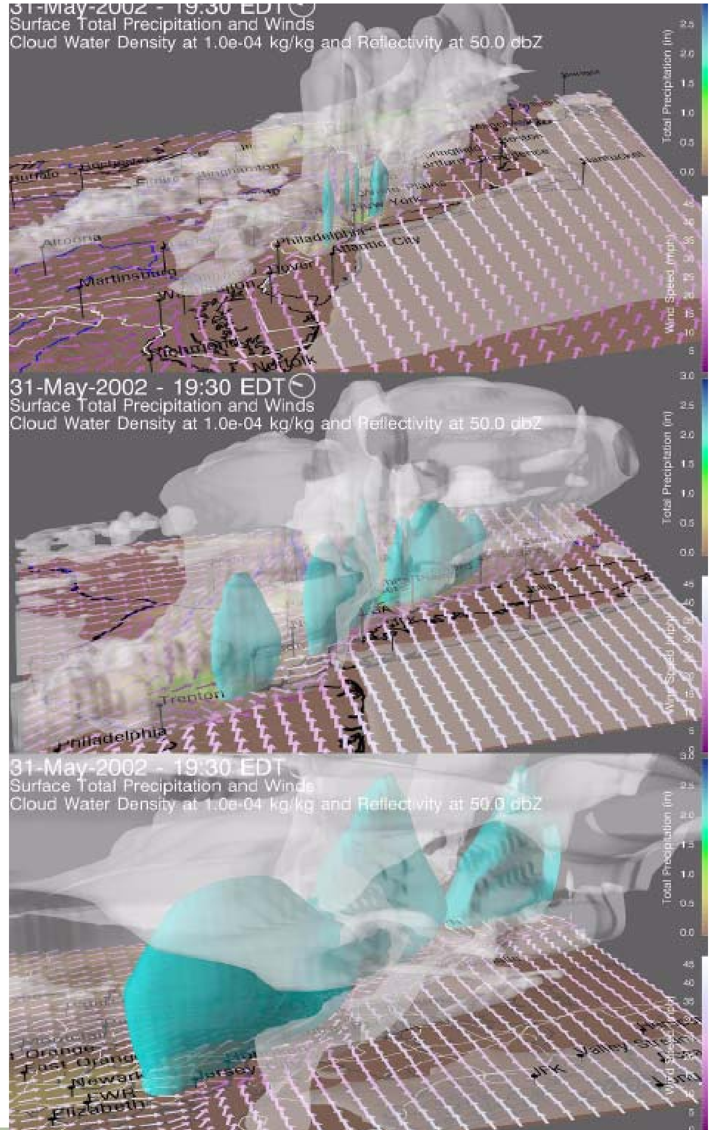
Modeling Wind Intermittency



The forecast for weather to be generated from Deep Thunder – 24-72 hr horizon

Deep Thunder – Forecasts for Weather-Sensitive Operations

- **Problem**: weather-sensitive business operations are often reactive to short-term (3 to 36 hours), local conditions (city, county, state) due to unavailability of appropriate predicted data at this scale
  - Energy, transportation, agriculture, insurance, broadcasting, sports, entertainment, tourism, construction, communications, emergency planning and security warnings
- **Solution**: application of reliable, affordable, weather models for predictive & proactive decision making & operational planning
  - Numerical weather forecasts coupled to business processes
  - Products and operations customized to business problems
  - Competitive advantage -- efficiency, safety, security and economic & societal benefit



Modeling Wind Intermittency

Scenario Reduction using Kantorovich Distance

- Consider two sets of sample paths
  - P = (P_i),= i = 1,…,n
  - Q = (Q_j), j = 1,…,m
  - where P_i=(p_i^1 ,…, p_i^T), Q_j=(q_j^1 ,…, q_j^T),
  - and P_i has probability $\bar{p}_i$, and Q_j has probability $\bar{q}_j$.
- Kantorovich distance between the two is defined as D(P,Q) = $\min_{x_{ij} \geq 0} \sum_{i,j} x_{ij} c_{ij}$

  s.t. $\sum_j x_{ij} = \bar{p}_i$

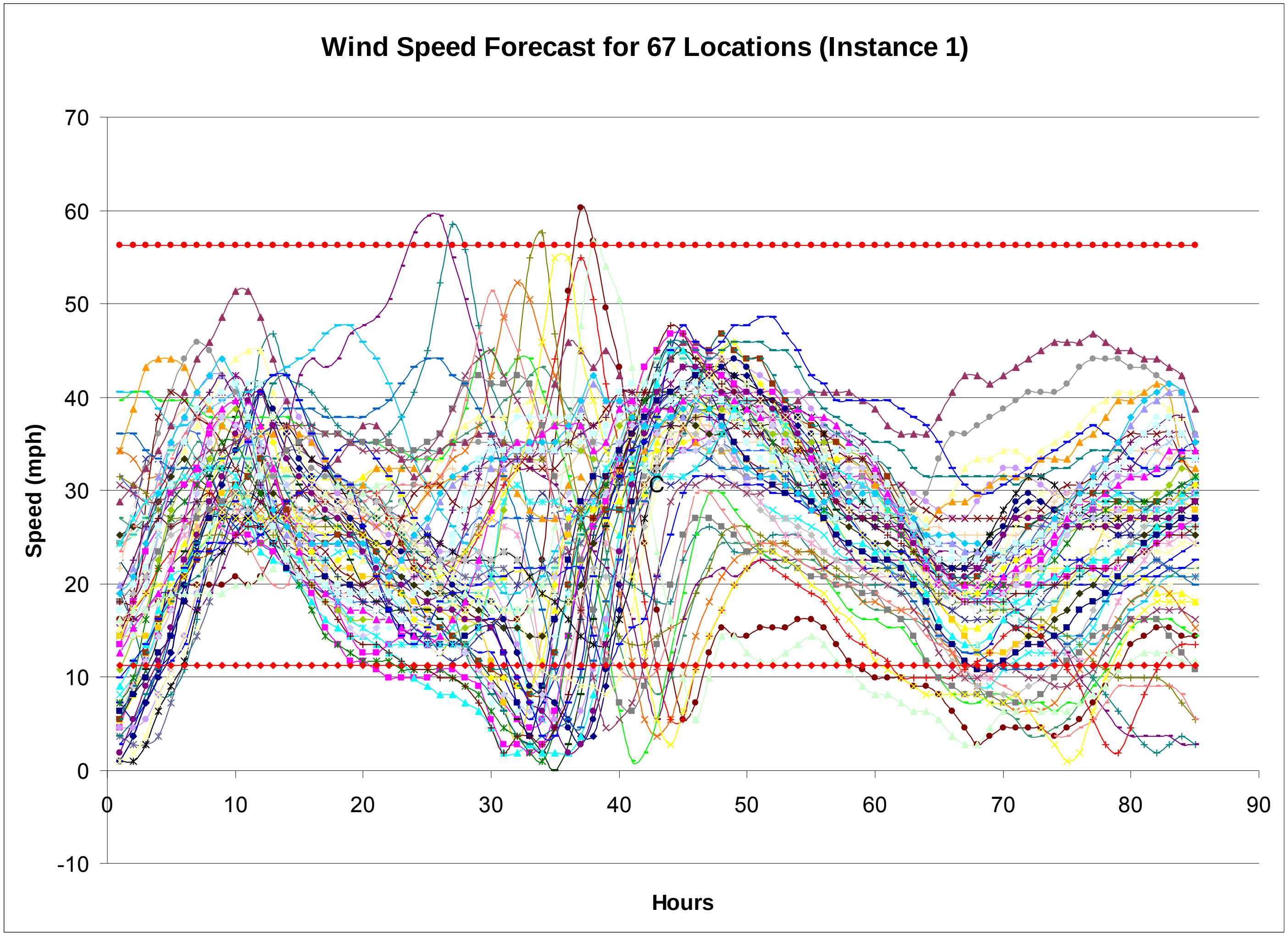
  $\sum_i x_{ij} = \bar{q}_j$

  where $c_{ij} = \sum_{t=1}^{T} | p_i^t - q_j^t |$.

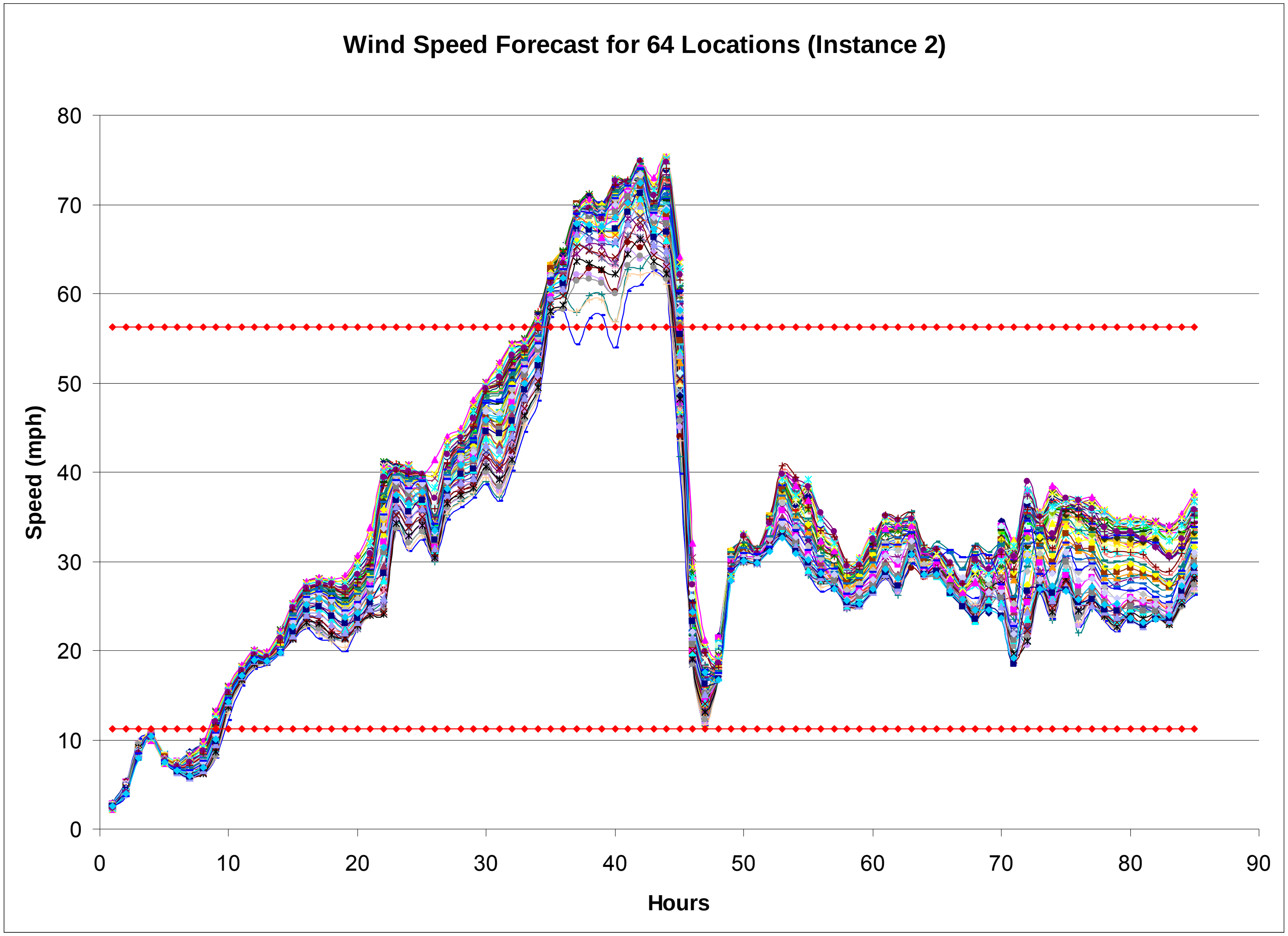
Stochastic Unit Commitment vs. Spinning Reserves

- Base demand is taken 2500 MWH for each 85 periods.
- Stochasticity is in the wind power. All of the wind power is used to meet the demand. Thus, the net demand to be met by the other units is stochastic.
- A wind-farm instance with 200 wind mills is considered.
- Using a scenario reduction technique, wind-power scenario tree is generated with 5, 10, 20 scenarios.

Wind speed forecasts –Instance 1

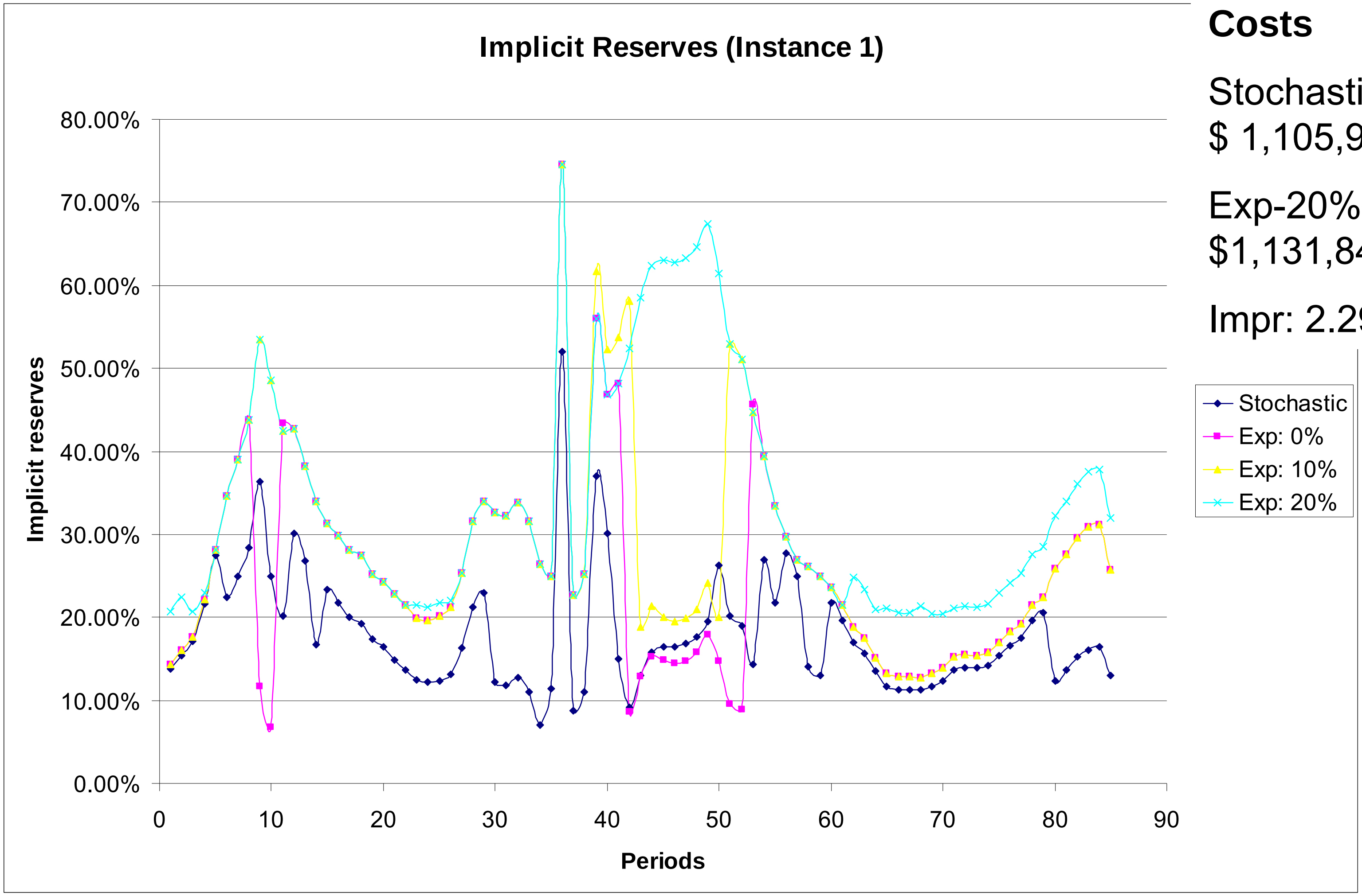


Wind speed forecasts –Instance 2



28

Implicit Reserves –Instance 1



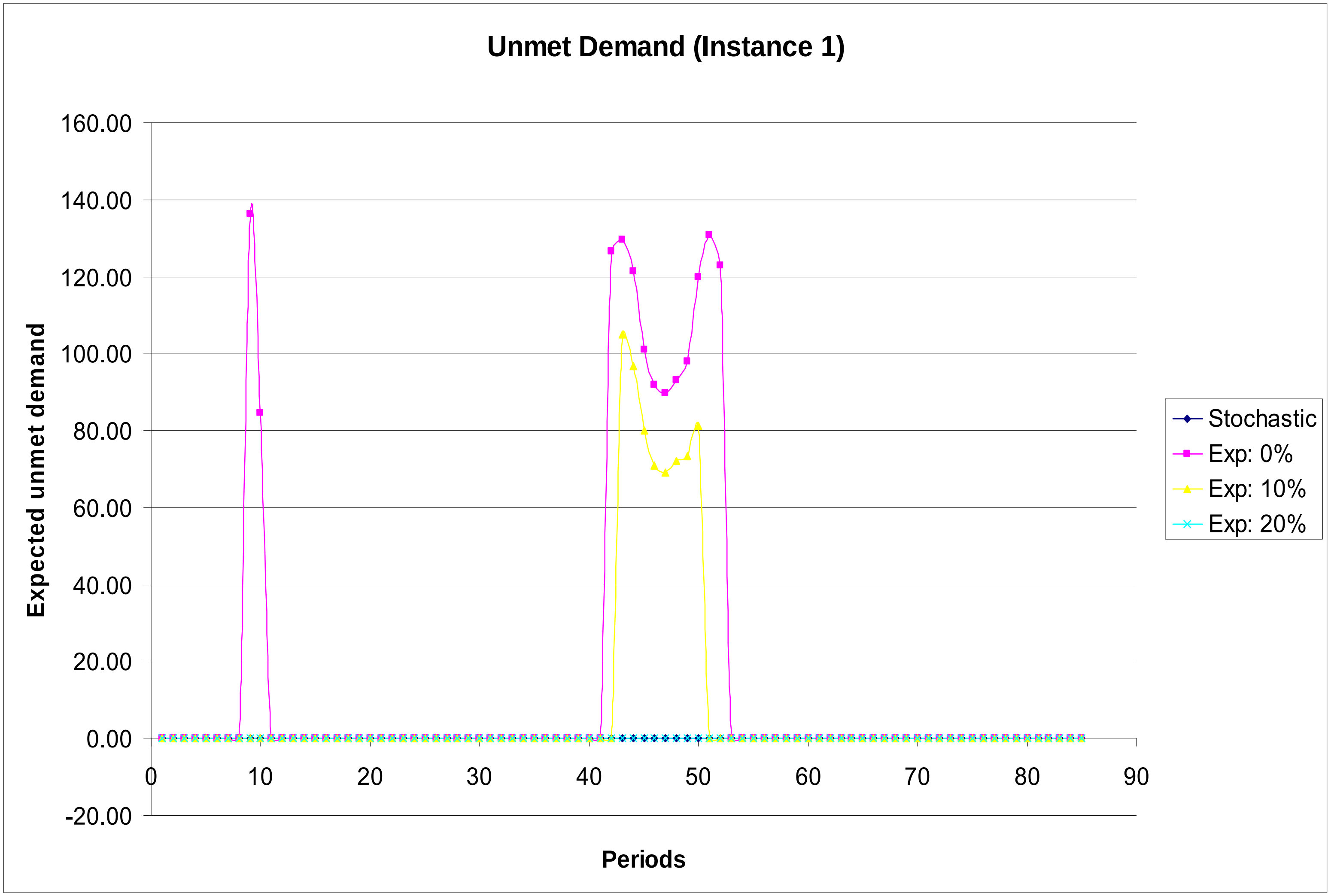
Costs

Stochastic:
$ 1,105,955

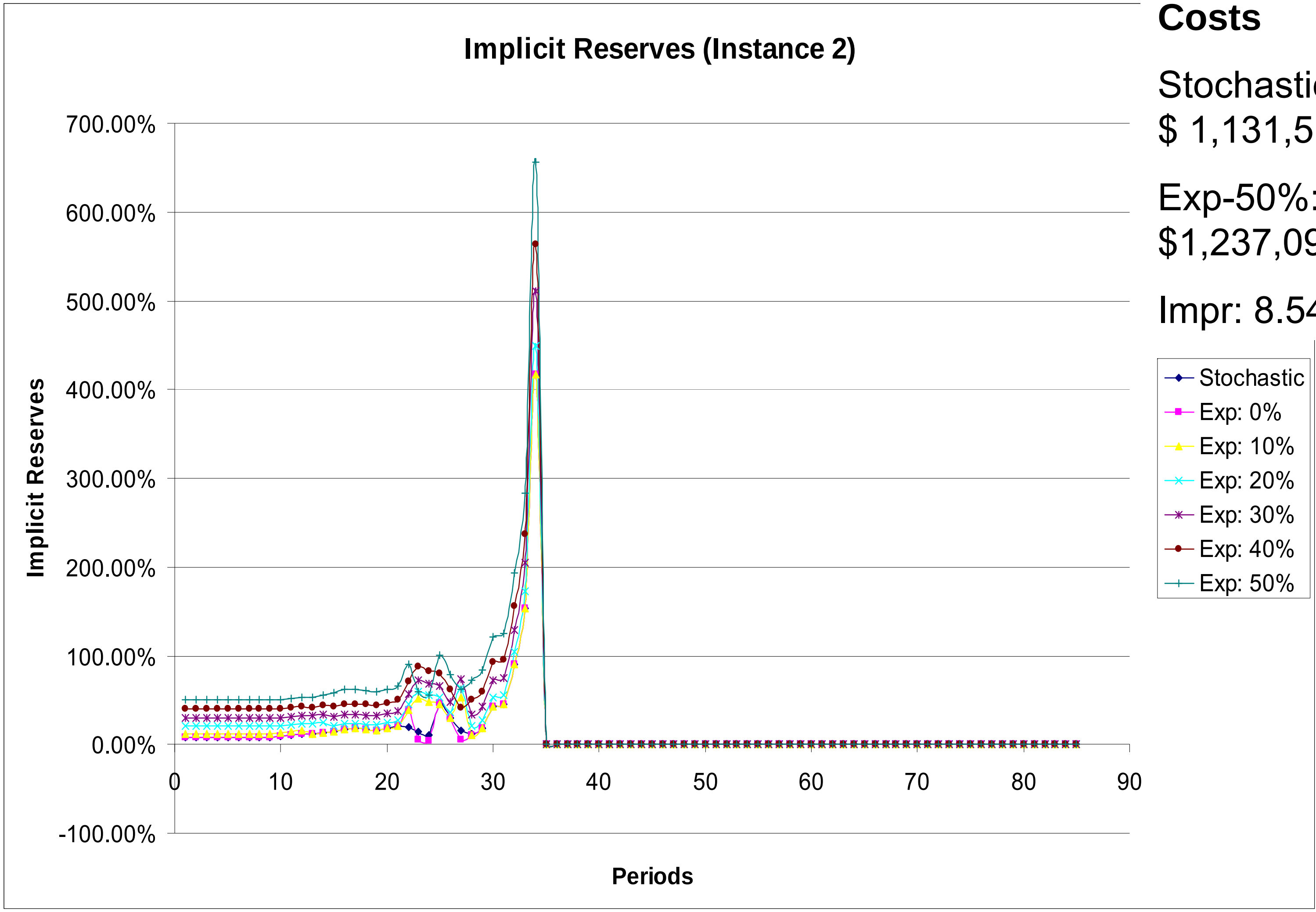
Exp-20%:
$1,131,847

Impr: 2.29%

Unmet Demand –Instance 1



Implicit Reserves –Instance 2



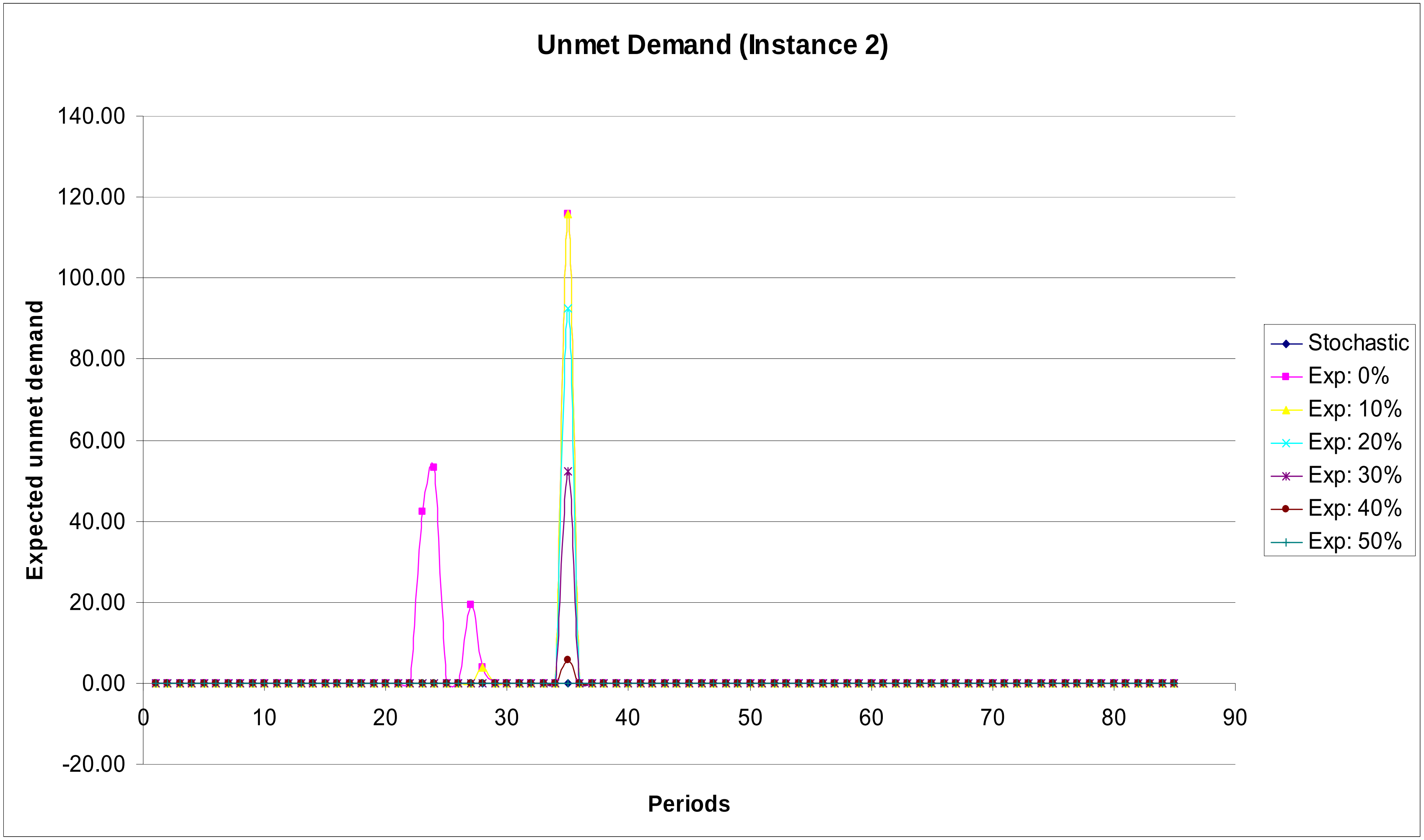
Costs

Stochastic:
$ 1,131,510

Exp-50%:
$1,237,099

Impr: 8.54%

Unmet Demand –Instance 2



Further Research

- **Real-life problems**
  - Handling power flows equations (linear vs non-linear)
  - Hundreds of units
  - Storage constraints
- **Scenario reduction**
- **Scaling the problem size**
  - Decomposition methods: branch-and-price
  - Parallel computing