Scalability and implementation issues in stochastic programming algorithms

Presented to the International Symposium on Mathematical Programming- Copenhagen,Denmark.

18-22 August, 2003

Chandra Poojari*, Frank Ellison*, Gautam Mitra*, Suvrajeet Sen#

*CARISMA: Centre for the analysis of risk and optimisation modelling applications

#University of Arizona

Acknowledgements: ***EPSRC(UK) supported Prof Sen's visit to CARISMA,***

Initiated as a result of a joint project commissioned by Unilever Research,

Outline

1. Background and Overview
2. Investigation framework
3. Benders decomposition
4. Stochastic decomposition
5. Experimental results
6. Discussions and Conclusions

OptiRisk SYSTEM

Background and Overview

Stochastic programming with recourse

$$\text{Min } \; c^{\mathrm{T}}x + \int p(\omega) f(\omega) y(\omega) \; d(\omega)$$

Subject to

$$Ax = b,$$
$$B(\omega)x + D(\omega)y(\omega) = h(\omega)$$
$$x \ge 0, y \ge 0.$$

ω	random event;
$p(\omega)$	Probability;
$f(\omega)$	Second-stage cost;
$B(\omega)$	Technical matrix;
$D(\omega)$	Recourse matrix;
$h(\omega)$	Right-hand side;
x	First-stage decisions;
$y(\omega)$	Second-stage decisions.

let $\xi_\omega = \{f, h, B, D\}_\omega$

$$\text{Min } \quad c^{\mathrm{T}}x + E_\xi Q(x, \xi)$$

Subject to $Ax = b,$

where

$$Q(x, \xi) = Min\ \{ fy(\omega) \mid Dy(\omega) = h - Bx,\ y(\omega) \ge 0\}$$

CARISMA

Background and Overview

Applications of SP with recourse models

1. Finance- Ziemba,Dempster,Pflug,Zenios,Mulvey,Van der Vlerk
2. Supply chain- Mitra, Eppen, Schrage
3. Transportation- Laporte, Louveaux,Powell,Cheung
4. Telecommunication- Sen,Gaivoronski
5. Airlines- Fergusson and Dantzig, Birge
6. Power- Pereira and Pinto, Takriti
7. Environment- Wallace,Gassmann

OptiRisk SYSTEMS

Background and Overview

Challenges in processing recourse models with discrete scenarios

$$\text{Min} \quad cx + \sum_{s=1}^{S} p_s f_s y_s$$

$$Ax \geq b$$

$$D_s y_s \geq h_s - B_s x$$

$$s = 1, \ldots, S$$

1. Piece-wise convex with a non-linear objective function.
2. Multi-dimensional integration where each integrand is a large LP.

CARISMA

OptiRisk SYSTEMS

Background and Overview

Systems for modelling and processing SP problems

Name	Affiliation	System Name
JJ Bisshop, et al.	Paragon Decision Technology	AIMMS
A Meeraus, et al.	GAMS	GAMS
R Fourer, et al.	Northwestern University	AMPL
MAH Dempster, et al.	Cambridge University	STOCHGEN
E Fragniere, et al.	University of Geneva	SETSTOCH
A King, et al.	IBM	OSL/SE
HI Gassmann, et al.	Dalhousie University	MSLiP
G Infanger et. al.	Stanford University	DECIS
P Kall, et al.	University of Zürich	SLP-IOR
G Mitra, et al.	Brunel University	SPInE

CARISMA

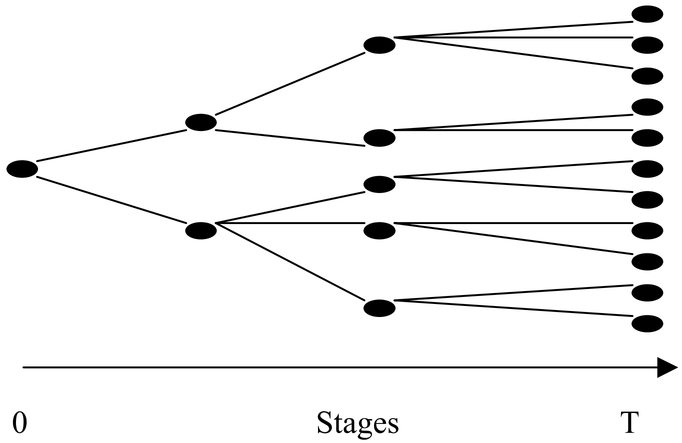
Background and Overview

Computational techniques for SP with recourse

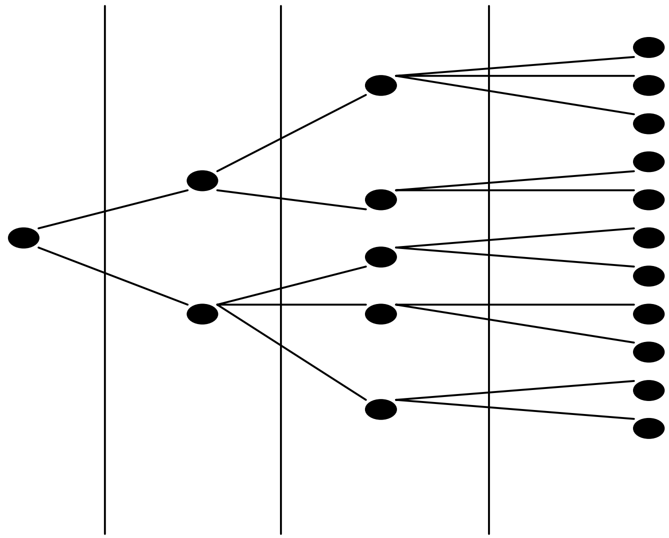
1. Direct methods (exploit structure)
   1. SSX- Kall, Strazicky, Fourer, Gondzio and Ruszcznyzski.
   2. IPM- Birge and Qi, Choi and Goldfarb, Gondzio et al., Hurd and Murphy.
2. Decomposition based methods
   1. Stage-wise - Van Slyke and Wets, Birge.
   2. Scenario – Rockafellar and Wets,Mulvey and Ruszcznyzski.
3. Sampling Based
   1. Shapiro, Infanger,Sen,Higle.

Background and Overview

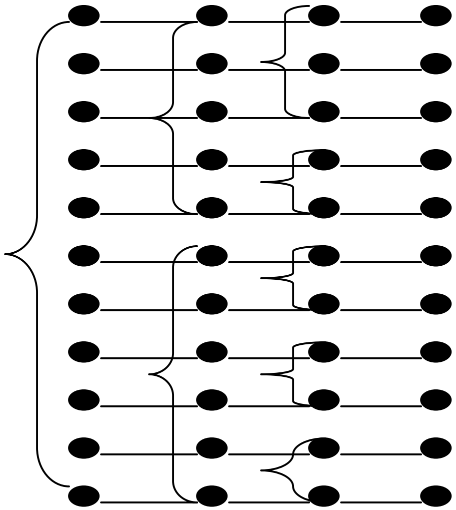
Decomposition based approaches



Stage-wise



Scenario



Investigation framework

The choice of the models were guided by

1. *Different application domains*
2. *Dimensions for scalability*

    2.1 Size of the expected value problem,

    2.2 Scenarios and stages.

OptiRisk SYSTEMS

Investigation framework

Model Instances …different domains

Model	Domain
20term	Electric capacity expansion
Pgp2	Electric capacity expansion
Storm	Scheduling
4node	Scheduling
AirL	Air lift operation
Clo125ms/ts	Supply chain
Fxm	Production scheduling
Ssn	Telecommunication
Phone	Telecommunication
Land	Electricity investment planning
Power	Power
Env	Energy and Environment planning
Pltexp	Capacity expansion

Acknowledgements

Andy Felt, John Birge and Derek Holmes, Suvrajeet Sen,
the CARISMA team

CARISMA

OptiRisk SYSTEMS

Investigation framework

Model Dimensions

Model	Stages	Scenarios	DEQ		
			Row	Cols	Non-Zeroes
20term	2	*	*	*	*
Pgp2	2	576	58	132	264
Storm	2	8/27/125/1000	528,185	1,259,121	3,341,696
4node	2	16/32/64/256	18,960	47,668	120,112
AirL	2	25	152	204	504
Clo125ts	2	125	1,216	3,040	7,822
Clo125ms	4	125	3,276	6,084	15,432
Fxm	4	216	386,940	517,282	4,518,039
Ssn	2	*	*	*	*
Phone	2	32,768	753,665	2,785,288	9,863,176
Land	2	3	23	40	92
Power	2	4	27	39	107
Env	2	5	288	294	852
Pltexp	6	7776	970,382	2,537,948	5,159,875

**: potentially infinite*

CARISMA

Benders decomposition- Algorithm

Step 1 *Cycle over all nodes in all time-stages from first to last*

1.1 Build sub-problem for current node

Apply parent solution-fix

Add any cuts stored for the node

1.2 Solve the sub-problem

1.3 If infeasible and if ***T=1*** *then: STOP (problem infeasible)*

1.5 If infeasible and ***T>1*** *then*

Build infeasibility-cut and store it for the parent node

Switch directly to the parent node and reset ***T = T-1***

Repeat from step 1.1

Step 2 *Cycle over all nodes from the first node in stage 2 to the last node*

2.1 Compute & Aggregate optimality cuts for stage ***T-1*** *parents*

2.2 When all child-nodes of one parent are complete:

Substitute parent solution in the aggregated cut to determine Lower bound for that subproblem at that node.

If either the parent has no optimality-cuts so far, or Lower bound ≠ Upper bound (within tolerance) then store aggregated cut for the parent node.

Step 3 *Cycle over all nodes in all time-stages*

3.1 Compare the objectives and θ-values of the node between current and previous iterations. Note if there are any differences

Step 4 *If both: cuts were created by step 2.2, and: there were no differences noted by step 3.1 then go back and repeat from step 1. Otherwise STOP (Optimum solution found)*

Benders decomposition – Data structure

Node Structure

1. Scenario, Time-stage and probability,
2. Link up the tree to the parent node,
3. Link down the tree to the first child node, together with child count enabling a cycle over all the child nodes from the current node,
4. Primal and Dual Solution vectors, Slack and Pi-vectors for the node sub-problem
5. Objective values, with and without Theta
6. Packed form of the most recent basis to use as a restart for the solver,
7. Indicators 'Solved' and 'Valid Basis',
8. Heading pointer to a list of optimality cuts,
9. Heading pointer to a list of feasibility cuts,
10. Counts of the number of cuts and other useful dimension-sizes

Cut Structure

1. The cuts are stored as a linked list,
2. Only the non-zeroes of the each cut are retained.

On going work in Benders

1. Regularisation of the master problem.
2. Warm-starts for the intermediate nodes in Nested Benders.
3. Experimentation with multi-cuts.

Stochastic decomposition- algorithm

Step 1 *Let* x^1 *be given*

Step 2 *Generate* $\xi(\omega)$
Solve the dual to get the cut coefficients
Update the previous cuts (hyperplanes)

Step 3 *Construct a new master using the hyper-planes generated so far*

Solve the master

Step 4 *Update the incumbent solution*

Step 5 *Construct and solve the `Primal regularised master'*

Stopping rules

-Using in-sample scenarios

-Using out-of sample scenarios

Stochastic decomposition- data structure

Node Structure

1. Incumbent solution
2. Dual vertices
3. Sampled scenarios

Cut Structure

1. Cuts stored as linked list.
2. Only the non-zeroes are stored.

On going work in Stochastic decomposition

1. Multi-stage SD.
2. Reduce the error in the statistical approximation through importance sampling.

Experimental Results

Computer specifications

- Model : Viglen Genie P4
- CPU : Pentium IV, 2.4 GHz
- RAM: 512 Mbytes
- OP system: Windows 2000

Approximation

For stochastic decomposition the multi-stage models have been aggregated to two-stage.

Comparison of Benders and Stochastic decomposition

Model	Benders	SD	
		Time(sec)	Iterations
20term		5	300
SSn	-	18	800
PltexpA5_16	-	41	950
PltexpA6_6	160	14	450
PltexpA6_16	-	301	2200
PltexpA7_6	-	12	1250
PltexpA7_16	-	96	1250
Stormg2_8	3	6	300
Stormg2_27	9	6	300
Stormg2_125	54	6	300
Stormg2_1000	542	6	300
Storm-I	-	7	300

Comparison of Benders and Stochastic decomposition

Model	Benders	SD
4node256	6610.12	6697
Fxm3_16	18439	18465
PltexpA4_16	-18.8510	-18.
PltexpA5_6	-23.2140	-23.2790
PltexpA6_6	-28.134	-28.1520
Stormg2_8	155352	154280

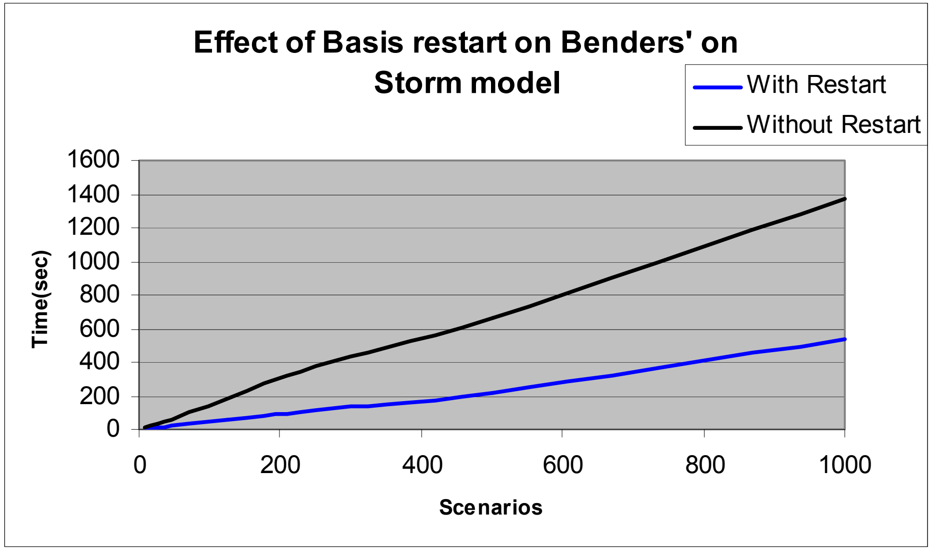
OptiRisk SYSTEMS

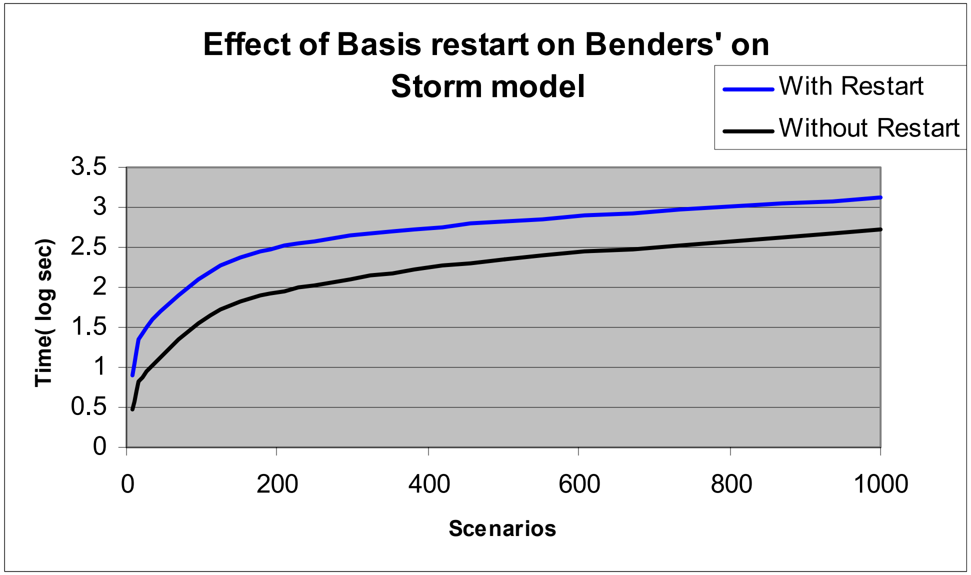
Effect of basis restart in Benders

Model	Benders	
	W/ Restart(sec)	W/O Restart(sec)
Fxm2_6	1	2
Fxm2_16	2	4
Fxm3_16	4	10
Fxm4_6	16	39
PltexpA4_16	46	59
PltexpA5_6	15	22
PltexpA6_6	160	316
Stormg2_8	3	8
Stormg2_27	9	31
Stormg2_125	54	187
Stormg2_250	109	376
Stormg2_500	220	663
Stormg2_1000	542	1370

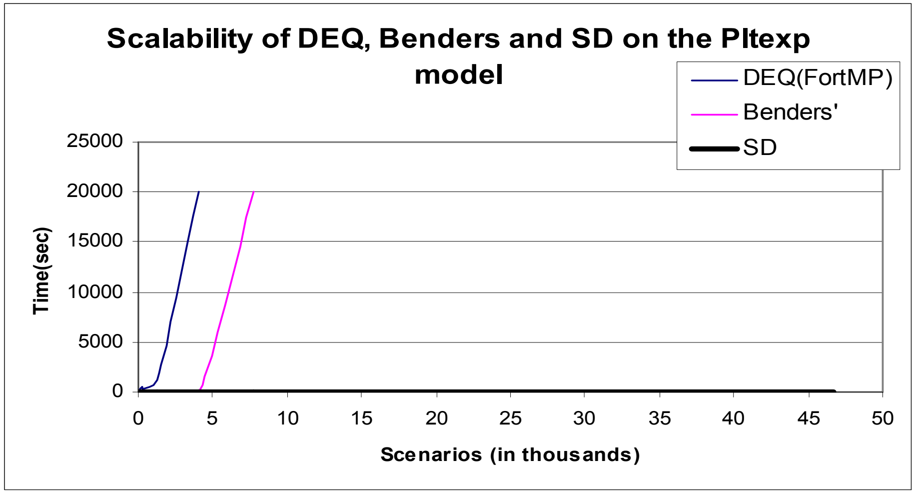
CARISMA

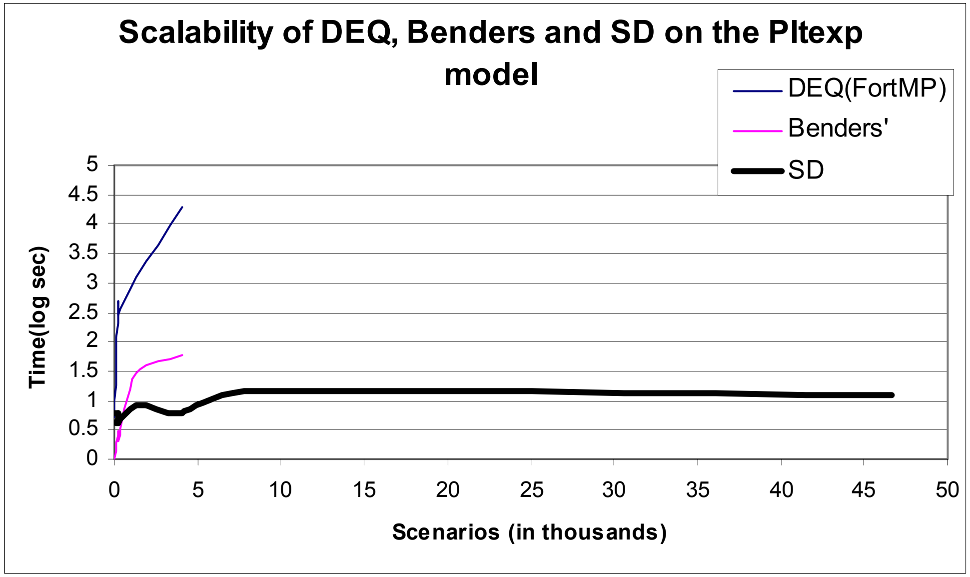
Basis restart in Benders Decomposition



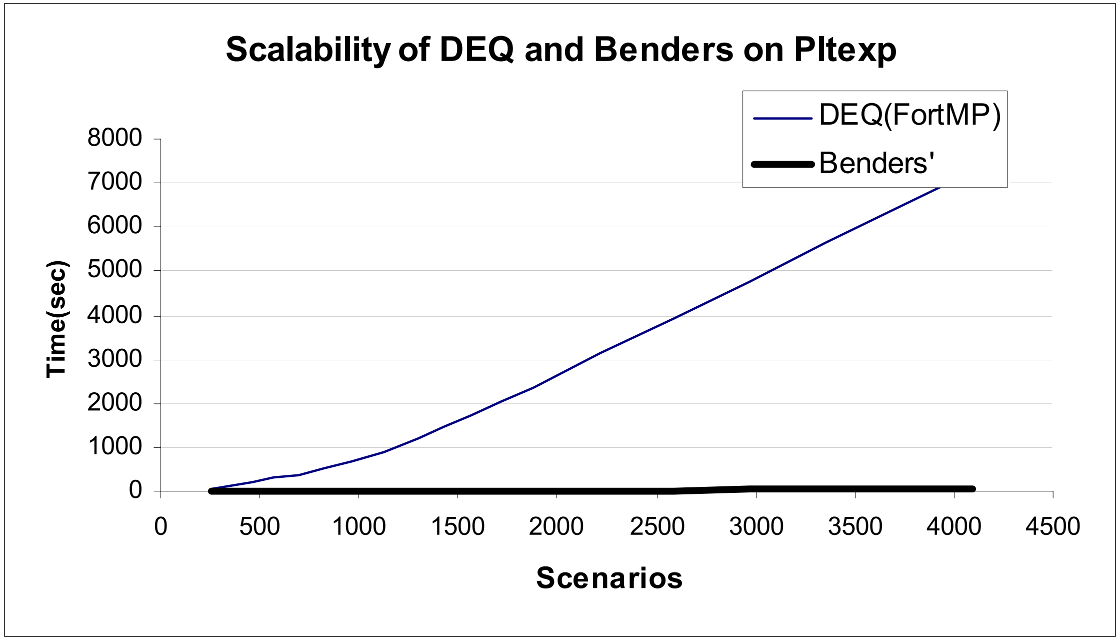


Scalability of the DEQ,Benders and SD on Pltexp model

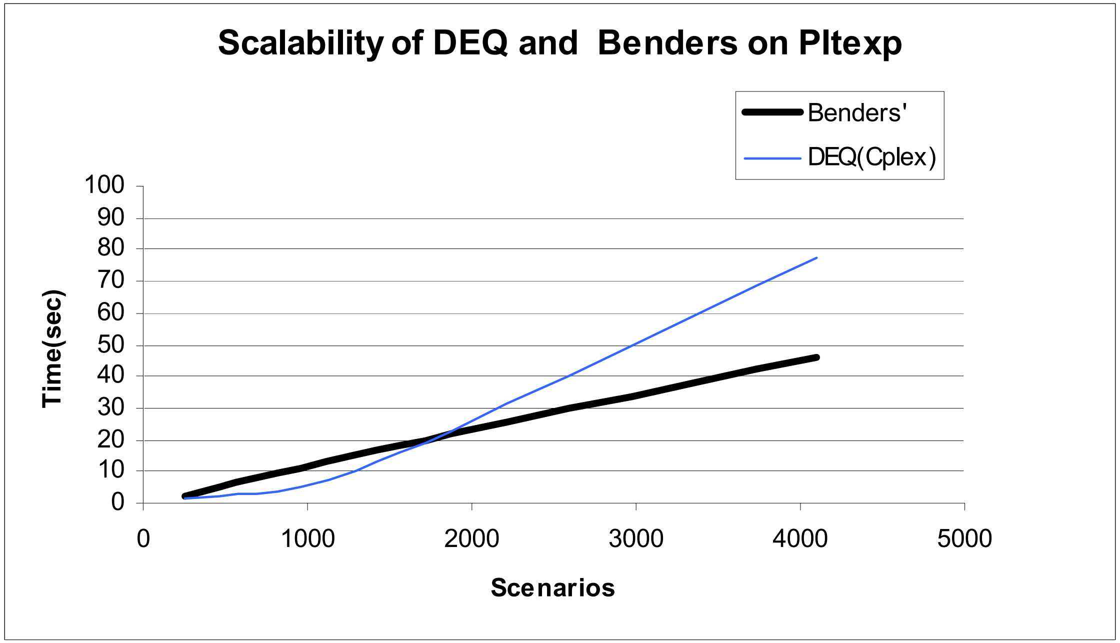




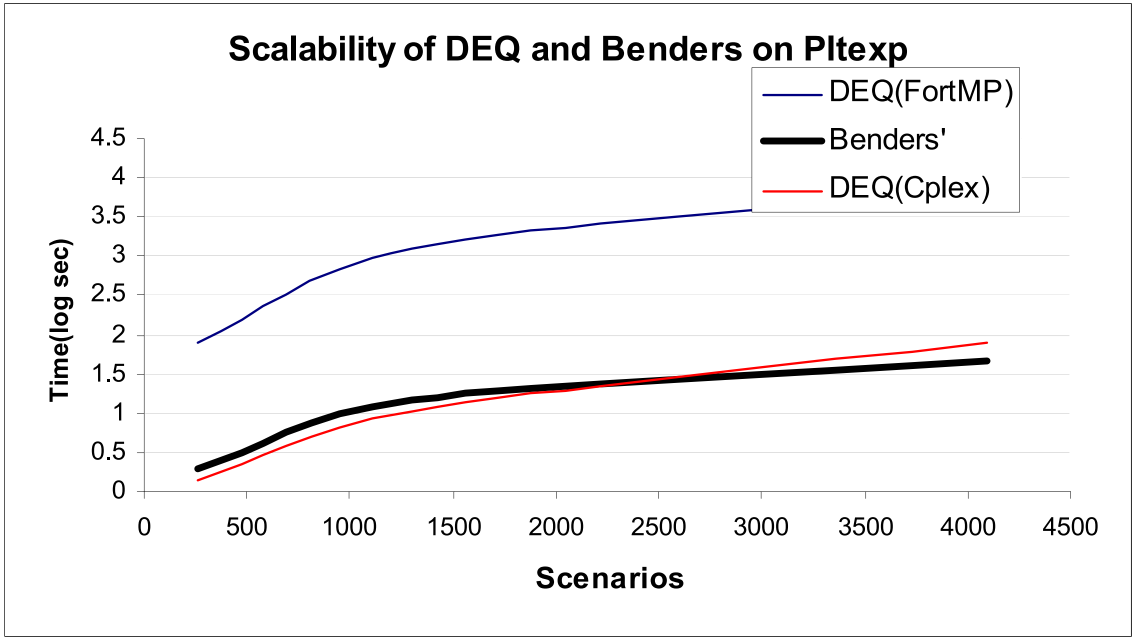
Scalability of DEQ and Benders on Pltexp



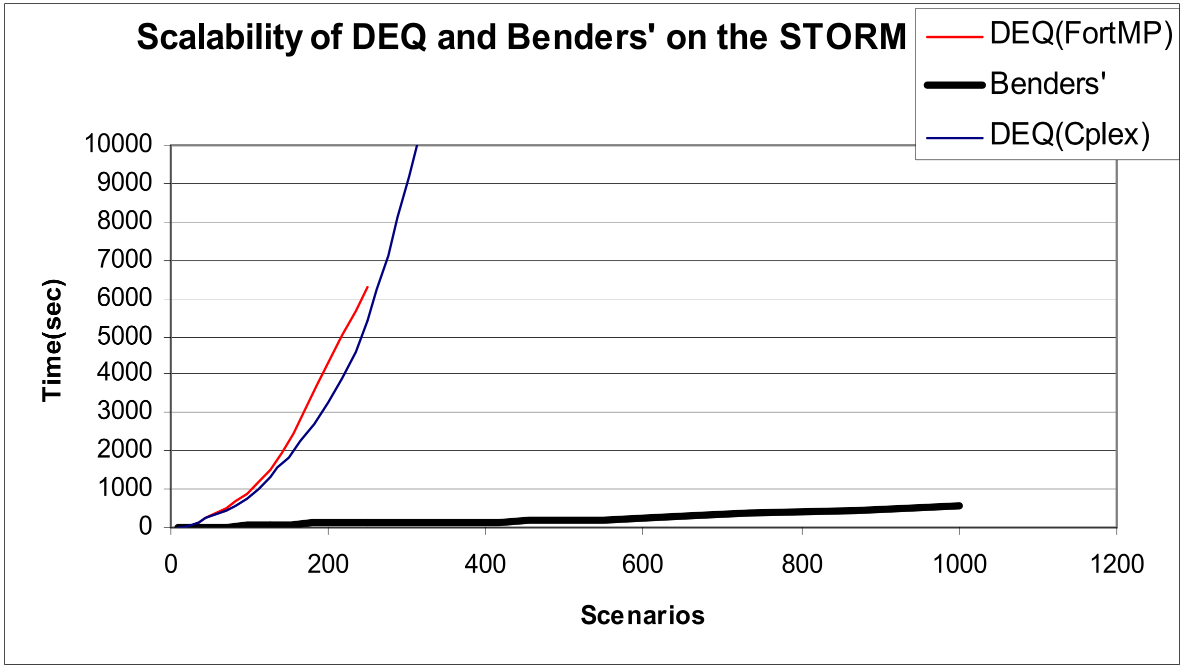
Scalability of DEQ and Benders on Pltexp



Scalability of DEQs and Benders on Pltexp

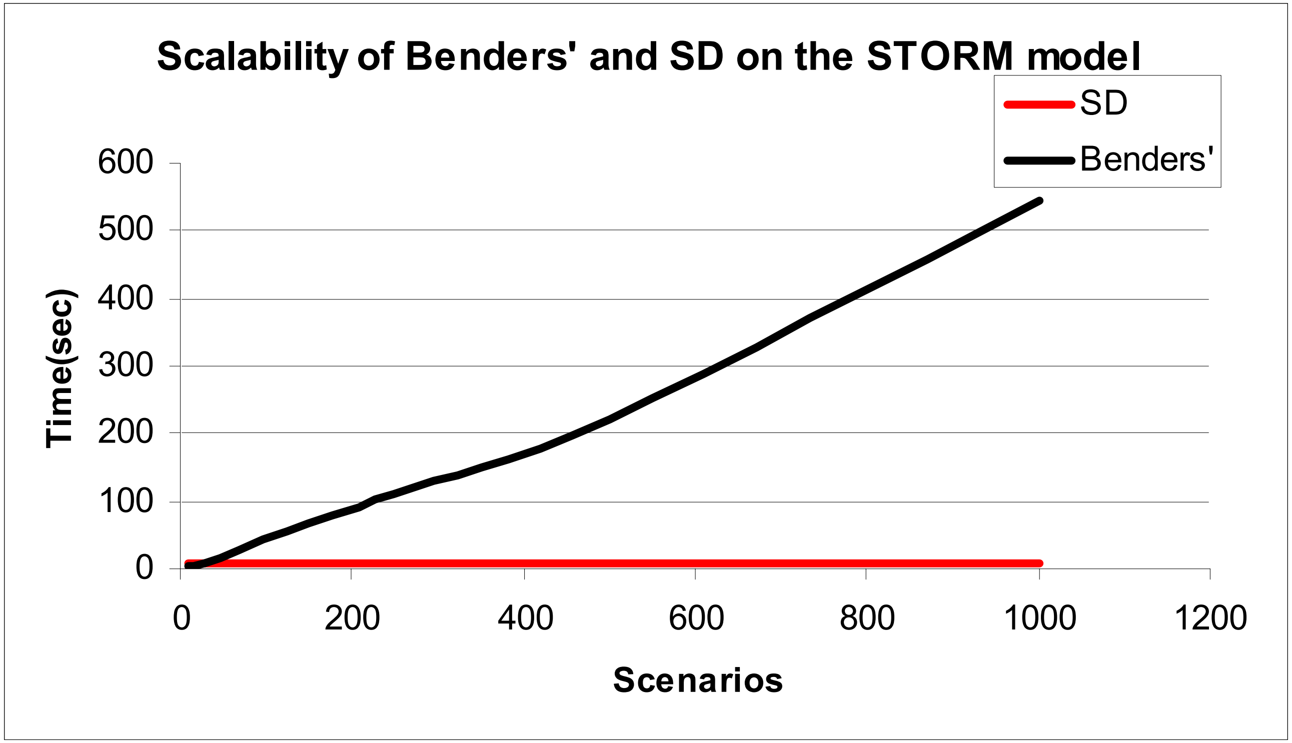


Scale-up of DEQs and Benders on the STORM model



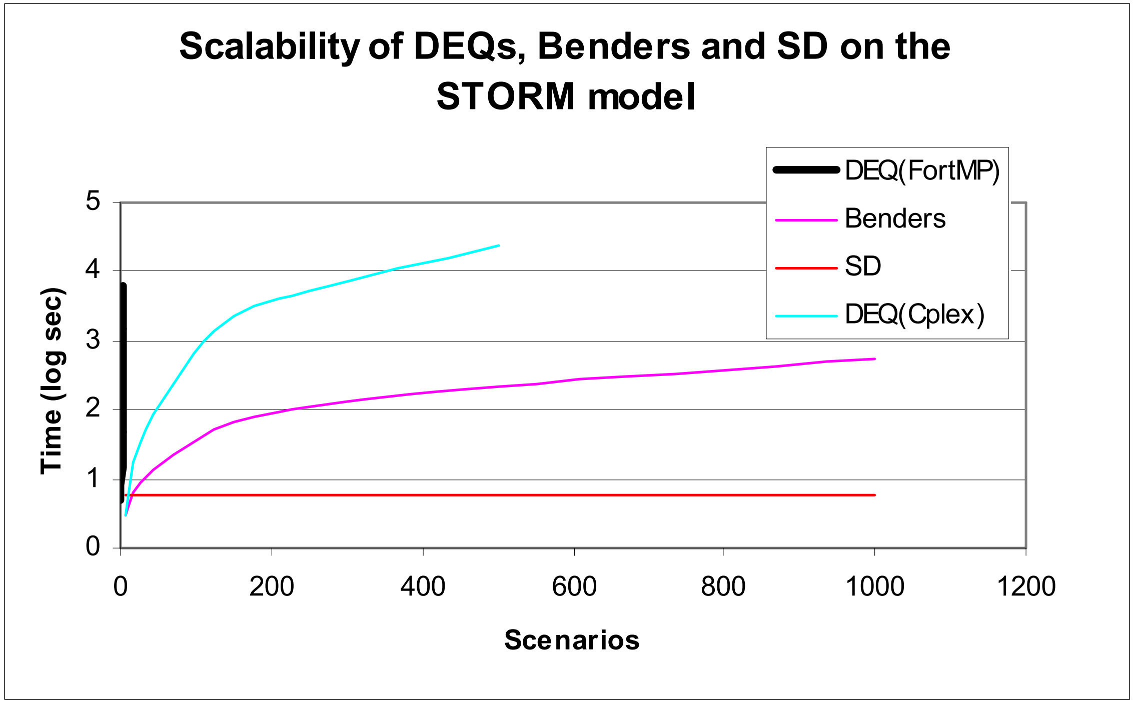
CARISMA

Scale-up of Benders and SD on the STORM model



CARISMA

Scale-up of Benders and SD on the STORM model



Discussions and Conclusions

1. Requests models in order The solvers (except SD) are available through
   1. NEOS
   2. OSP (www.osp-craft.com)
   3. WEBOPT(www.webopt.org)
2. A stochastic programming integrated environment (SPInE) is available to academics- contact ***software@optirisk-systems.com***.
3. Parallelisation of Benders and Stochastic decomposition
4. Enhance the test sets for SP – help of the SP community required.
5. Future work: PHA, DQA.
6. Algorithms for processing Stochastic Integer programming problems

Thank you
and have a nice time at Copenhagen

Email: **Chandra.Poojari**@brunel.ac.uk

http://www.optirisk-systems.com/

http://carisma.brunel.ac.uk/

Thank you
and have a nice time at Copenhagen

URL: http://www.carisma.brunel.ac.uk

Email: Chandra.Poojari@brunel.ac.uk