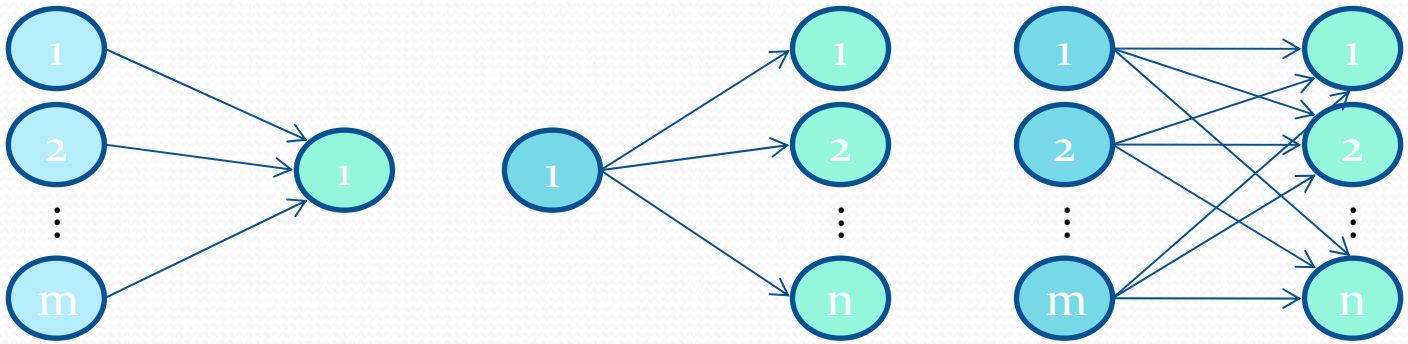
Mechanism Design Auctions

Vadim Ovtchinnikov
Mentor: Professor John Ledyard
SURF 2009

What is an Auction?

- Derived from the Latin word augere and means “to increase”.
- Set of trading rules between buyers and sellers, created to determine a winner that increases the flow of wealth through the auction.



Combinatorial: Bids on TV spots advertisement “A&B&C or D&E&F or…”

My work on this project!

Generated bids in Python
Optimization processes in AMPL/CPLEX
Computed the averages and made distribution plots of Gains from trades, Buyers surplus and Sellers surplus and compare them to Maximum possible gains from trades.

Goal is to identify auction design that yields highest Gains from trade.

Important designs

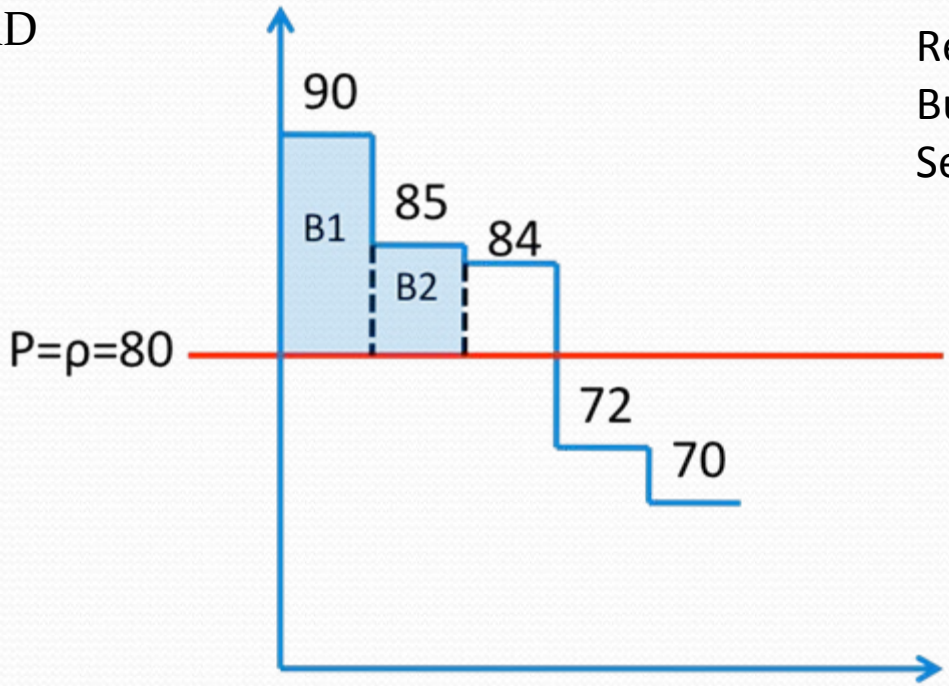
- Sealed bid auction (RAD- Resource Allocation Design)

- Ascending bid auction (CC- Combinatorial Clock)

Simple environment, Example 1

Example 1: Simple (homogenous commodities)	
One seller, M = 1	Five buyers, J = 5
Q(1) = 2 ρ (1)= 80	β =[70,90,84,85,72]^T B[j]=100 for all j=1,2,…,5 Demand[j]=1 for all j=1,2,…,5

RAD

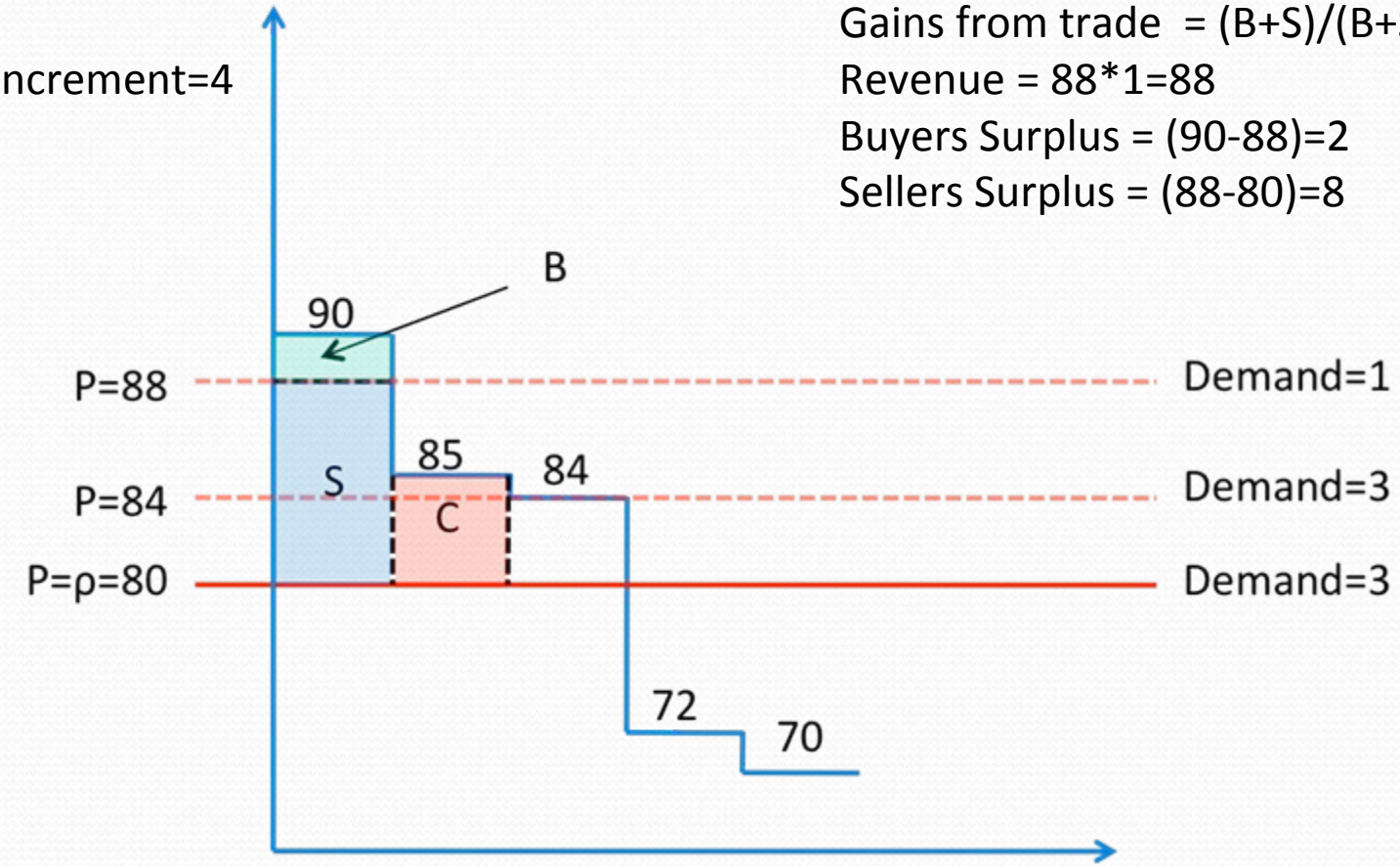


Revenue = 80*1+80*1=160
Buyers Surplus = (90-80)+(85-80)=15
Sellers Surplus = (80-80)+(80-80)=0

Simple environment, Example 1

CC,
Price increment=4

Gains from trade = (B+S)/(B+S+C)
Revenue = 88*1=88
Buyers Surplus = (90-88)=2
Sellers Surplus = (88-80)=8



Generation of simple environment

Simple environment	
Fix parameters:	Number of sellers (M=1) and buyers (J=5); B[j]=100 for all j; Q[1]=2; Demand[j] = 1 for all j;
Randomly generated parameters:	ρ on [0,100] β on [75,200]

Simple environment, simulation:

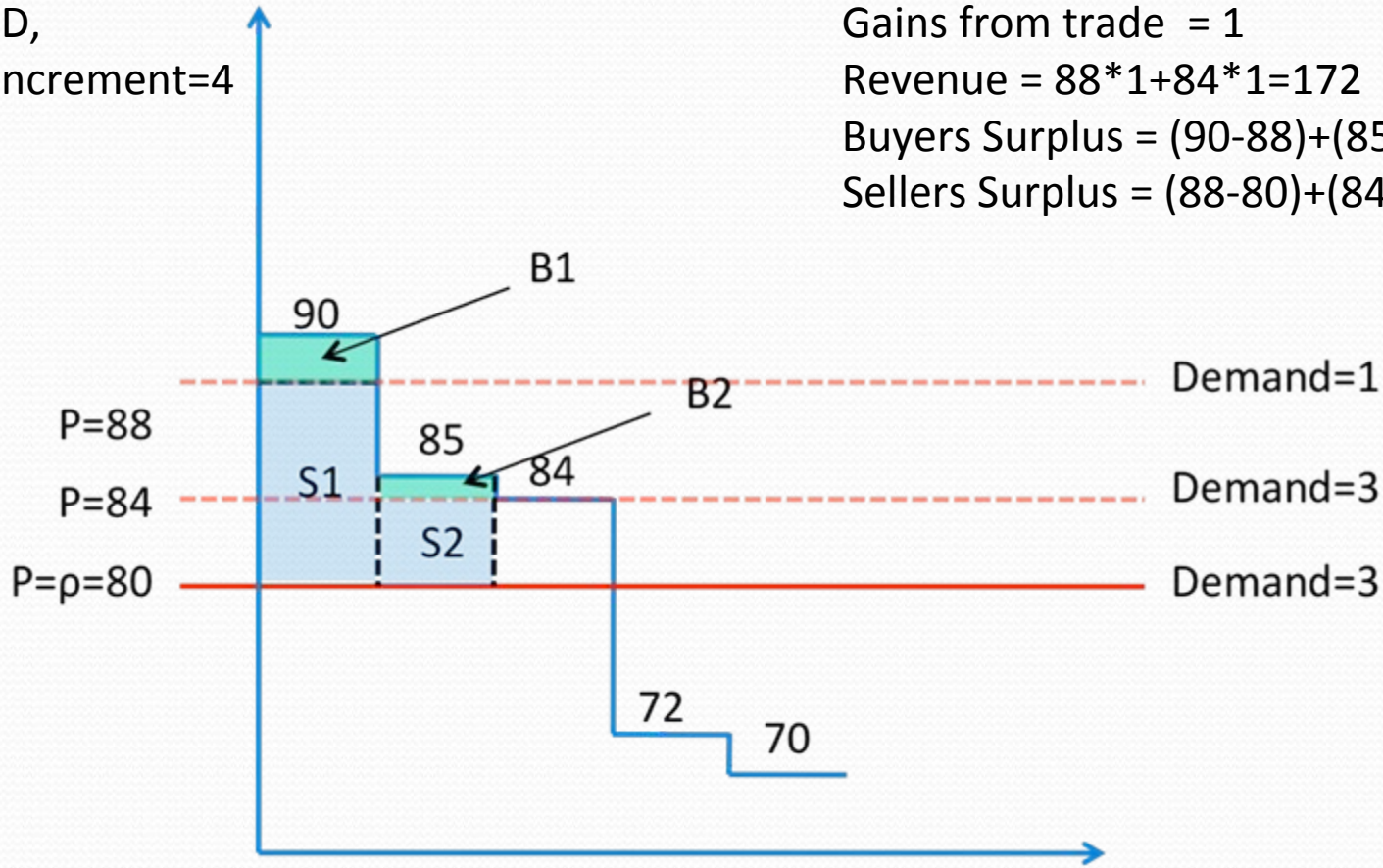
	Gains From Trade	Revenue	Buyers Surplus	Sellers Surplus	Number of iterations
RAD	14.0 %	12.8 %	14.0 %	0.0 %	1.00
CC	3.5 %	3.8 %	1.9 %	1.5 %	12.67
CC+WD	72.4%	68.2%	38.4%	34.0%	13.59
RAD+CC+WD	74.8%	69.8%	39.2%	35.6%	7.97

Does not work! Fix up the computer auction by introducing Winners Determination (WD).
Uses “past bids” as environment, and generates allocation by maximizing sellers surplus.
Note: Every input in the table is divided by Maximum G.F.T.
Revenue = sum{j,i}(P[j,i]*x[j,i])
RAD: Chooses a smarter starting and CC+WD runs from there.

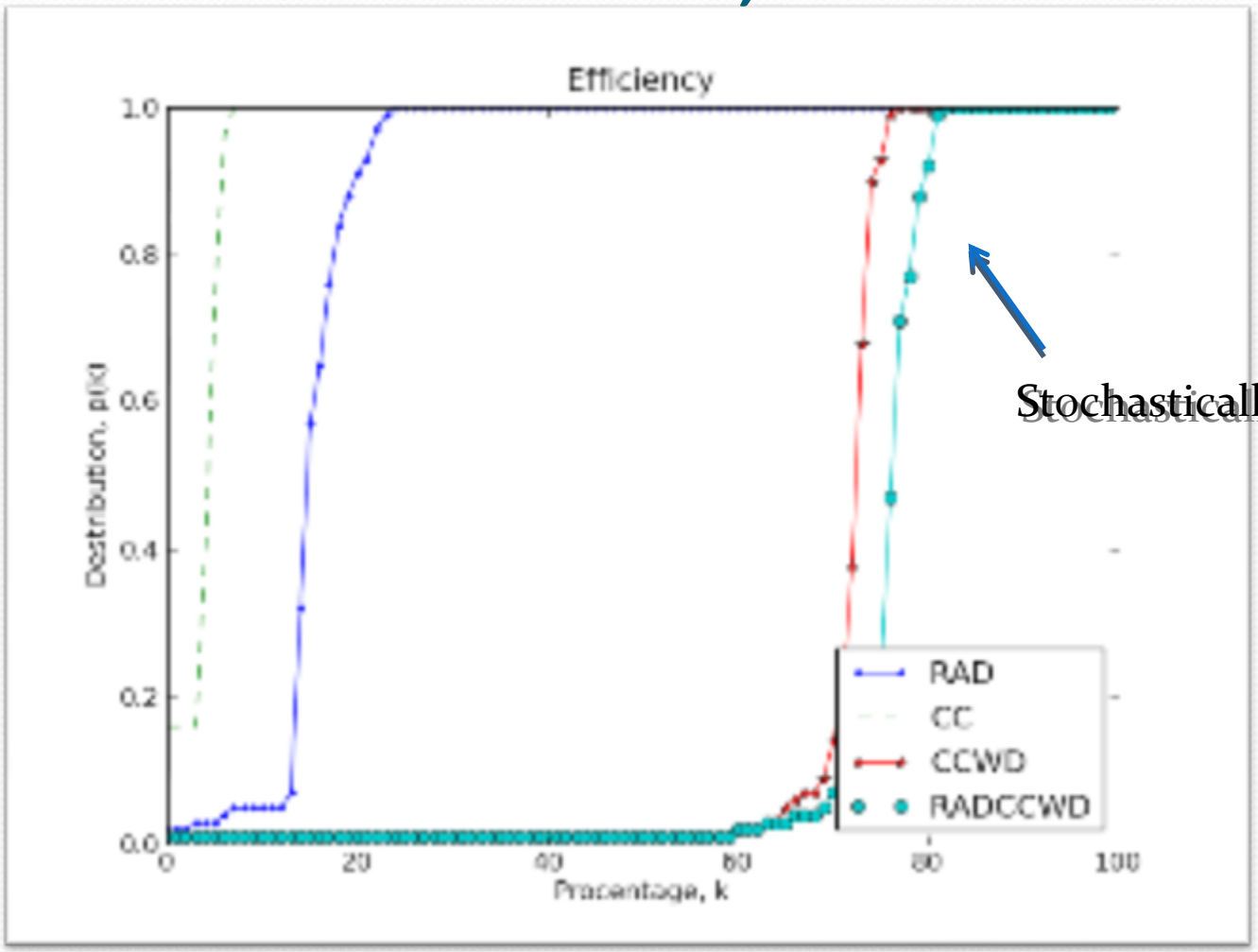
Simple environment, Example 1

CC+WD,
Price increment=4

Gains from trade = 1
Revenue = 88*1+84*1=172
Buyers Surplus = (90-88)+(85-84)=3
Sellers Surplus = (88-80)+(84-80)=12



Simple environment, simulation:



Complex environment:

Environment for TV advertisement.

E4

	Slots	Impressions /Slot	price
program 1	105	56	550
program 2	90	75	630
program 3	75	78	650
program 4	60	81	800

	P1	P2	P3	P4	Budget	Max s/p
Program Buyer 1	448	600	624	648	$ 25,000	43
Program Buyer 2	672	900	936	972	$ 37,000	64

	Max Spots	Max/spot	Budget	Max s/p
Spot Buyer 1	52	810	$ 30,000	52
Spot Buyer 2	62	891	$ 36,000	62

	Max Impressions	Max CPM	Budget
Impression Buyer 1	3,500	9	$ 28,000
Impression Buyer 2	4,750	12	$ 38,000

We randomly generated all this numbers, in reasonable ranges and ran the auctions for each draw.

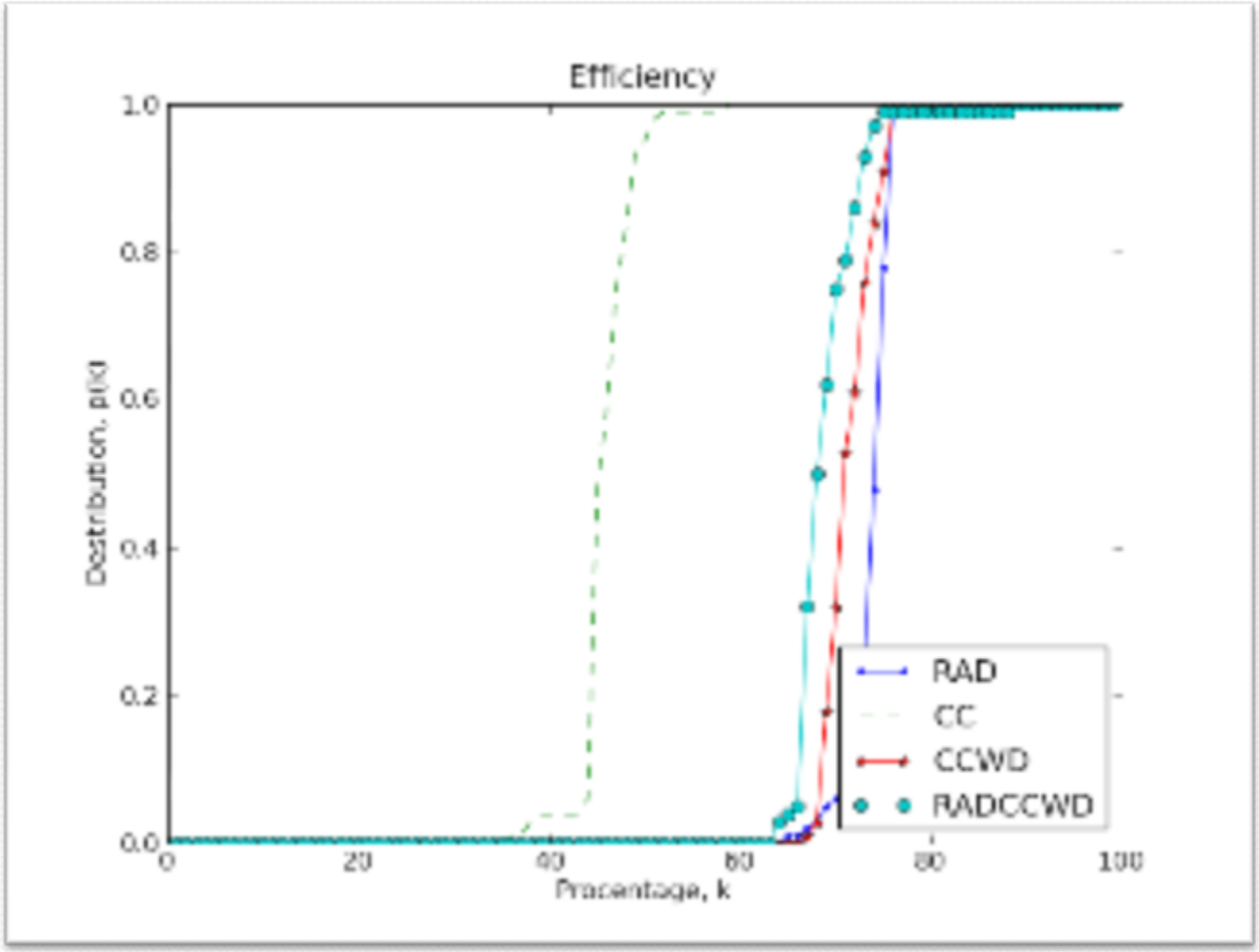
Complex environment: Simulation:

	Gains From Trade	Revenue	Buyers Surplus	Sellers Surplus	Number of iterations
RAD	73.2 %	133.9 %	73.2%	0.0 %	1.00
CC	44.2 %	106.6 %	18.1 %	26.0 %	9.74
CC+WD	68.7 %	175.7 %	21.2 %	47.5 %	10.46
RAD+CC+WD	66.5 %	155.9 %	20.6 %	45.9 %	3.35

RAD has o% Sellers Surplus, in the long run sellers want show up for this auction.

RAD+CC+WD have almost the same values as CC+WD but runs in fewer iterations.

Complex environment: Simulation:



Conclusion

RAD+CC+WD, clearly best for simple environment!

RAD+CC+WD give good values per number of iterations for the complex environment!

Open question, are there better auctions designs?

Acknowledgments

Professor John Ledyard, co-mentor Guilherme de Freitas and the SURF program!