STOCHASTIC PROGRAMMING:

Algorithmic Challenges

Plenary Lecture at the
The Nineth International SP Conference
(August 28, 2001)

Suvrajeet Sen
SIE Department,University of Arizona
Tucson, AZ

Collaborators: Partners in Crime

Today's talk is based on work with several individuals, especially my long-time colleague Julia Higle (AZ). Also:

Michael Casey (AZ)
Guglielmo Lulli (Italy and AZ)
Lewis Ntaimo (AZ)
Brenda Rayco (Belgium and AZ)
Yijia Xu (AZ)
Lihua Yu (AZ)

This Presentation: Transitions from Continuous to Discrete

1. Lessons from successful algorithms

- Convexity and Decomposition
- Special structure
- Sampling
- Inexact "solves"

2. "Informal" exploration of challenges in multi-stage problems

- Scenario trees, stopping criteria and estimates of solution quality
- Real-time Algorithms
- Multi-granularity multi-stage models

3. “Less Informal” exploration into Stochastic IP

- Literature
- Two Stage SIP: Stochastic Polyhedral Combinatorics
- Multi-stage SIP

4. Conclusions

1. Lessons from Successful Algorithms (for Continuous Problems)

1.1 Convexity and Decomposition:

- Benders' Decomposition (L-shaped Method), and its extensions to Regularized, Stochastic and Interior Point methods provide resource directive decomposition-coordination approaches.

Work of Birge, Dantzig, Gassmann, Goffin, Higle, Ruszczynski, Sen, Vial, Wets and others.

Convexity of the value functions provides the justification

- Scenario Aggregation/Decomposition provides a certain price directive (Augmented Lagrangian-type) approach.

Work of Rockafellar, Ruszczynski, Wets and others.

Duality and hence convexity again provides the basis

1.2 Special structure: Stochastic linear programming

Polyhedral structure of the value function of LPs help streamline computations.

It is well known that for problems with finite support (i.e. finitely many scenarios), Benders' decomposition is finite. This is also true for regularized decomposition (see work of Kiwiel, Ruszczynski)

Homem de Mello and Shapiro show that sampling also leads to an optimal solution in finitely many steps (for SLP with finite support).

Work with Higle shows how the Stochastic Decomposition method by-passes LP "solves" by a matrix update for fixed recourse problems

1.3 Sampling: Large number of scenarios

$$\min_{x \in X} \quad f(x) := E[h(x, \tilde{\omega})]$$

- Since $f(x)$ is difficult to evaluate, algorithmic schemes replace $f(x)$ by $f_k(x)$, where k is an iteration counter.
- For deterministic algorithms f_k are obtained by the same deterministic selection of scenarios: $\{\omega_t\}_{t=1}^{N}$

For stochastic algorithms f_k are obtained by sampling scenarios

- Stochastic-Quasi Gradients: Work of Ermoliev, Gaivoronski, Uryasiev etc.
- Successive Sample Mean Optimization (Stochastic Counterpart/Sample Average Approximation, “Retrospective Optimization” in Simulation).
- The approach: create one sample mean function, optimize it; create another sample mean function (with larger sample size), optimize that, and so on. This is really a “meta”-concept:

- each sample mean optimization is a SP
- does not use information generated in one iteration for subsequent ones.

- Stochastic Decomposition approximates the sample mean function by one “cut” in each iteration and each “cut” progressively approximates a sample mean function resulting from increasing the sample size.
- Common random numbers reduce variance, and allow recursive updates.

- Sampling in Multi-stage Problems: By solving a dual SLP, one can use a stochastic cutting plane algorithm (a la SD). This algorithm, called SSD, will be discussed in detail in Brenda Rayco's presentation. A brief observation though ... aggregation techniques can reduce the growth of the master dramtically.

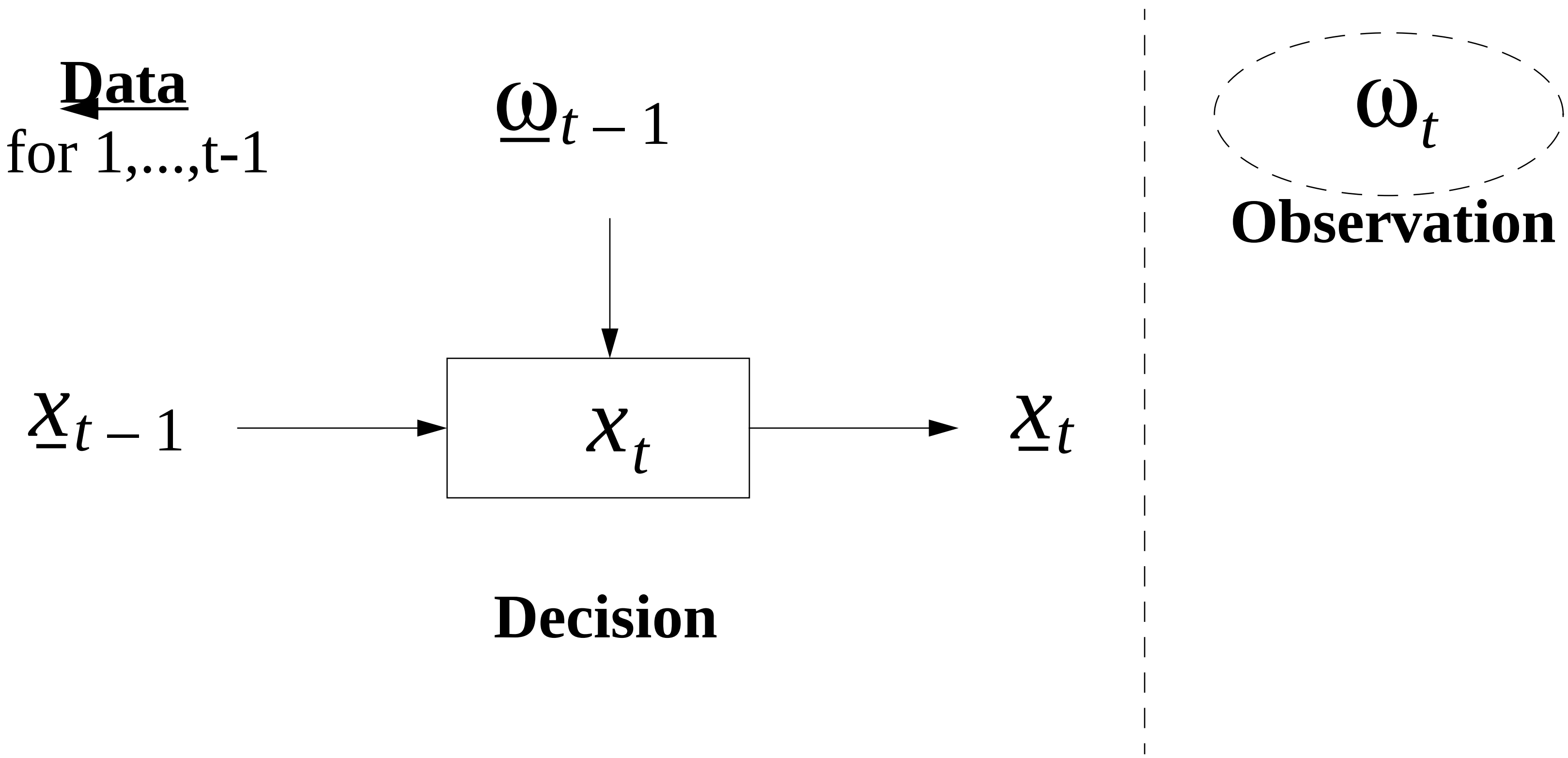
1.4 Inexact "solves"

- Not as common in SP as in Nonlinear Programming and Integer Programming
- In SP, the Scenario Aggregation method allows inexact solves, but implementations have typically not used this feature.
- The "argmax" procedure in SD provides "inexact solves"

- A recent version of Benders' decomposition, known as Abridged Benders Decomposition (Work of Birge and his students) allows inexact solves in subproblems

This feature is extremely important for SIP algorithms since the subproblems are IP.

2. "Informal" Exploration of Challenges for Multi-stage SP



- For $t = 2,\ldots,T$, define functions.

$$f_t(\underline{x}_{t-1}, \underline{\omega}_{t-1}) = \text{Min } c_t(x_t, \underline{x}_{t-1}, \underline{\omega}_{t-1}) + E[f_{t+1}(\underline{x}_t, \tilde{\omega}_t \mid \underline{\omega}_{t-1})]$$
$$\text{s.t. } x_t \in X_t(\underline{x}_{t-1}, \underline{\omega}_{t-1})$$

Assuming $f_{T+1} = 0$, the decision problem is

$$\text{Min } c_1(x_1) + E[f_2(x_1, \tilde{\omega}_1)]$$
$$x_1 \in X_1$$

2.1 Scenario Trees

- Current approaches seek a discrete approximation (of a given size) which satisfy some properties associated with the stochastic process. (The work of Consigli, Dempster, Dupacova, Hoyland, Mulvey, Wallace)
- Pflug develops a nonlinear optimization problem which seeks the “nearest” scenario tree of a specified size which provides the best approximation.

- How could one develop a sequence of trees (of the stochastic process) which provide solutions with certain guarantees? Frauen-dorfer's Barycentric method provides a partial answer.
- Approximations using probability metrics (for problems with finite support) appears to be promising (Dupacova, Growe-Kuska, Romisch)

Suppose one approximates the original SP using some discrete approximation. What is the quality of the resulting first stage solution? ... Ouput Analysis

2.2 Real-time Algorithms

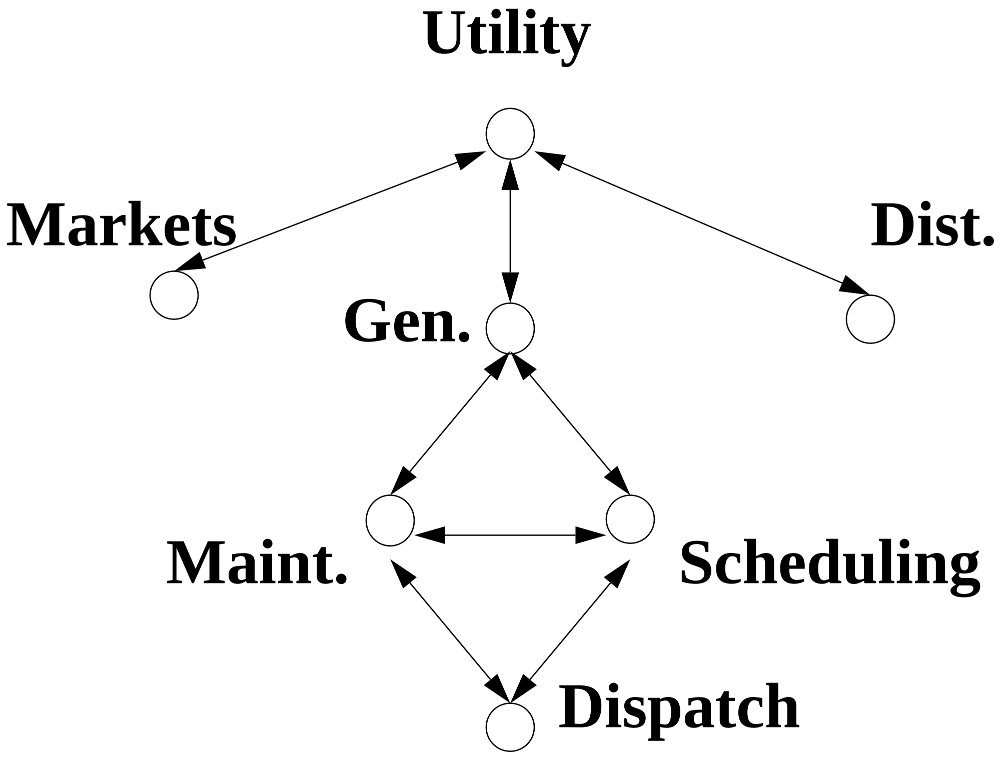
- "Nested simple recourse problems."

  - Recourse decisions in real-time problems must be made within constraints of computational time.

  - Models consist of multi-stage "simple-recourse decisions." Such "trajectory planning models" may warrant continuous random variables (e.g. wind speed at future locations) on the right-hand side.

- Real-time decision and control problems
  - Example: a collection of mobile automatons know their own location, but only know approximate locations of others.
  - Location information is updated with the passage of time. Collision-free path planning problems lead to multi-stage real-time scheduling problems.
  - In AZ with Ntaimo and Xu. Similar applications by W. Powell, A. Kleywegt et al.

2.3 Multi-granularity Multi-stage Models



- Decisions of one group affects operations of another.
- Modeling time-lags is important. In our example, power contracts agreed to in month t, will affect production in month $t+s$.
- Each group may have decision-aids that capture a particular time-scale quite well. For instance, dispatching decisions may be daily, generation (unit commitment) plan-

ning may happen week-by-week, power contracts may range from “day ahead” to “six-months” ahead.

How should we coordinate such decisions?

(Work with Lulli, Yu, and an AZ power company)

3. “Less Informal” Exploration into SIP: The Transition to Discrete Problems

- Our view is based on successes for continuous problems ... successful algorithms for SIP problems will ultimately use
  - Convexity and Decomposition
  - Special structure
  - Inexact “solves”
  - Sampling for Large Scale Problems

3.1 Literature

- Two stage simple integer recourse
  Series of papers by Klein-Haneveld, Stougie and van der Vlerk (well solved)
- Two stage 0-1 Problems
  Laporte and Louveaux
- Two stage General Integer Problems
  Schultz, Stougie and van der Vlerk
  Hemmeke and Schultz (SPEPS)

- Cutting planes for two stage problems
  Caroe (dissertation)
  Caroe and Tind
  Ahmed, Tawarmalani, Sahinidis (SPEPS)
  Sen and Higle (SPEPS)
  Sherali and Fraticelli
- Multi-stage Problems
  Birge and Dempster (see also Sen, Higle, and Birge)
  Lokketangen and Woodruff
  Caroe and Schultz

3.1 Two Stage Problems: Stochastic Polyhedral Combinatorics

What role does polyhedral combinatorics play in deterministic IP?

- Reduces size of the search tree in B&B

One should expect the stochastic versions (of cuts) to play the same role

Consider the following 2-stage SIP

$$
\begin{array}{lll}
\text{Min} & c^T x + \Sigma_s \; p_s g_s^T y_s & \\
\text{s.t} & Ax & \geq b \\
 & T_s x + W y_s & \geq \omega_s \;\; \text{for all s} \\
 & x \in Z_+^{n_1}, \;\; y_s \in Z_+^{n_2} &
\end{array}
$$

- Caroe's approach:

  a) Solve SLP relaxation of SIP;

  b) If solution is integer, stop;

  c) Else, develop a "cut" for each non-integer pair (x, y_s).

  d) Update the SIP by adding cuts

  - Repeat from a)
- Observations:

  - Note the close connection with Deterministic IP

- Various different cuts are permitted (Gomory, “Lift-and-Project” etc.)

- Each cut involves only (x, y_s). Thus L-shaped structure of SIP is maintained and the SLP relaxation can be solved using L-shaped method.

- Caroe suggests “lift-and-project” cuts for binary problems.

- Disjunctive Decomposition (D^2) Algorithm (with Higle):
- Decompose the problem into two stages.

  a) Given a first stage *x*, solve an LP relaxation of the second, and strengthen each second stage convexification whenever necessary. If no further strengthening is necessary, stop.

  b) Convexify the value function approximation of each second stage IP.

  c) Update the master program; repeat.

- Observations: Special structure (Linear inequalities, fixed recourse) allows all scenarios to share common cut-coefficients (C^3 Theorem).
- Cut generation is simple recourse LP

- Does not reduce to a known IP method for problems with only one deterministic scenario, that is, this is also a new IP decomposition method.

Convergence for 0-1 Stochastic MILP

Assumptions

- Complete recourse
- All second stage integer variables are 0-1
- First stage feasibility requires extreme points of X as in 0-1 problems
- Maintain all cuts in W^k
- If there are multiple "fractional variables" to form a disjunction, choose one with

smallest index, and recall the matrix from the most recent iteration at which the same variable was used for cut formation

Under these assumptions, the D^2 method results in a convergent algorithm.

Extensions to allow Branch-and-Cut and continuous first stage decisions are currently underway. (Work with Higle and Sherali)

3.2 Multi-stage SIP

Even more important to use special structure for realistic problems

For examples of the use of special structure, see papers by Takriti, Birge and Long; Nowak and Romisch.

Very few general algorithms available for this class of problems to date.

- Caroe and Schultz propose a Branch-and-Bound method in which bounds are calculated using Lagrangian relaxation
  - Dual iterates generated with Kiwiel's NDO algorithm
  - Computations are reported for 2-stage problems, although the development is valid for multi-stage problems.

- Several important advantages ...

- Two stage and multi-stage problems handled with equal ease
- It is possible to take advantage of special structure
- Transition from deterministic to stochastic model is easy

- Branch and Price for MSIP (Work with Lulli)

Motivation

- Has many of the same advantages associ-ated with Lagrangian Relaxation
  - Handles 2-stage and Multi-stage SIP
  - Allows exploitation of special structure
- Makes greater use of LP software (mature)
  - Warm starts are easy to handle
  - Sensitivity analysis is routine
  - Greater availability of reliable code

For notational simplicity consider a two-stage problem

$$\begin{aligned}
\text{Min} \quad & c^T x + \Sigma_s \; p_s g_s^T y_s \\
\text{s.t} \quad & Ax \geq b \\
& T_s x + W_s y_s \geq \omega_s \quad \text{for all s} \\
& x \in Z_+^{n_1}, \;\; y_s \in Z_+^{n_2}
\end{aligned}$$

We have chosen a two stage, pure integer problem only for notational ease. Solving multi-stage, and mixed integer problems add no additional conceptual complexity ... only greater computational work.

The general Branch-and-Price idea is to have a master IP that enforces non-anticipativity, and the subproblems are deterministic multi-stage problems.

Both master and sub-problems enforce integer restrictions

For each scenario "s", subproblems generate integer points $(x_{s,r}, y_{s,r})$, where r is an index associated with an integer point. Let

$$f_{s,r} = c^T x_{s,r} + g_s^T y_{s,r}$$

- As in "column generation" schemes, each of these points will be associated with a "column" in the master program

- The rows in the master program will consist of
  - First stage constraints (optional)
  - Non-anticipativity constraints
  - Convexity constraint
  - Bounds on x’s used in branching
  - Bounds on y’s used in branching

The master problem at any B&B node is:

$$\max_{x,\,\alpha} \quad \Sigma_s \Sigma_r \; p_s f_{s,r} \alpha_{s,r}$$

$$\text{s.t} \quad Ax \leq b$$

$$x - \Sigma_s \Sigma_r p_s x_{s,r} \alpha_{s,r} = 0$$

$$\Sigma_r \alpha_{s,r} = 1, \quad \text{for all s}$$

$l_j \leq x_j \leq u_j$, j among x branches,

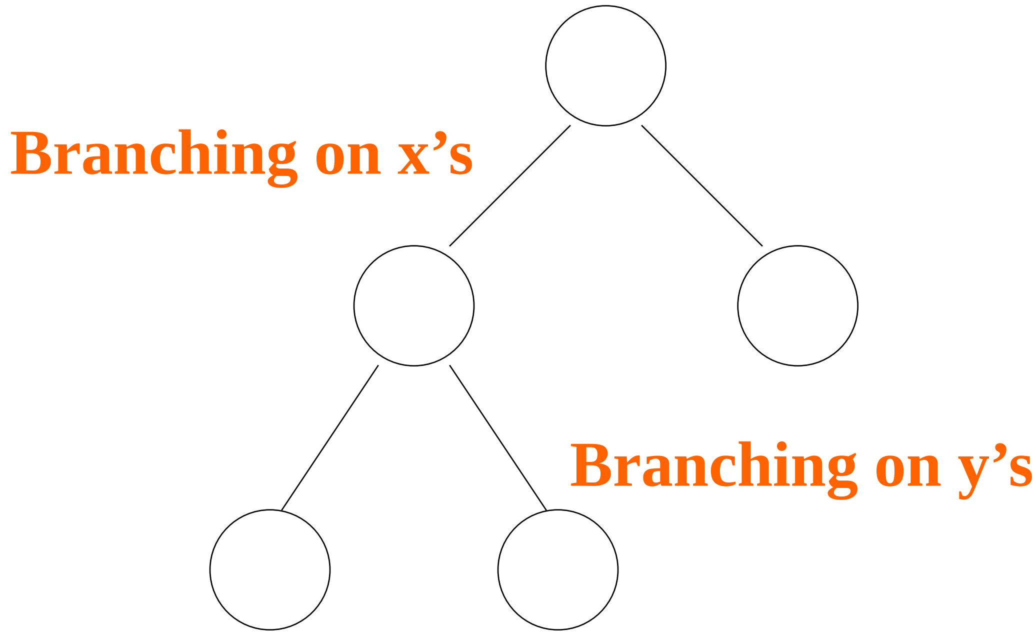
$L_{s,i} \leq \Sigma_r y_{s,r,i} \alpha_{s,r} \leq U_{s,i}$, i among y branches

The Basic Scheme

- For any node q of the B&B tree, solve the nodal problem using column generation.
- If x_j^q is the value of variable x_j for node q and, this value is fractional, then this variable is a candidate for branching.
- Similarly, if for some scenario s, the value

$$\Sigma_r y_{s,\,r,\,i} \alpha_{s,\,r}^q$$

is fractional, then we may use this to generate two new nodes of a B&B tree.



Master Program B&B Tree

- The pricing problem for scenario *s* has the form

$$
\begin{aligned}
\text{Min} \quad & \hat{c}_s^T x + \hat{g}_s^T y_s \\
\text{s.t} \quad & T_s x + W_s y_s \qquad \geq \omega_s \\
& x \in Z_+^{n_1}, \quad y_s \in Z_+^{n_2}
\end{aligned}
$$

- Note that this problem maintains the special structure that may be associated with a scenario problem. Thus, if we're interested in solving Stochastic Dynamic Lot Sizing Problems, each pricing problem is a Dynamic Deterministic Lot Sizing Problem

- Also, each pricing problem can be solved in parallel. (These advantages are the same as in Lagrangian Relaxation)

3.3 Computations for Multi-stage SIP (Work with G. Lulli)

Branch-and-Price concepts were applied to a batch sizing problem ... an extension of dynamic lot sizing problems. In such problems, one studies trade-offs between production/setup costs with inventory holding costs.

Assuming no backlogging, or probabilistic constraints, the stochastic batch sizing model is written as follows

$$
\begin{aligned}
\text{Min} \quad & \Sigma_s \; p_s \; \Sigma_t \; c_t x_{ts} + f_t y_{ts} + h_t I_{ts} \\
\text{s.t.} \quad & I_{ts} = I_{t-1,\,s} + b x_{ts} - d_{ts} \\
& x_{ts} \leq M_t y_{ts} \\
& (x_{ts}, I_{ts}) \geq 0 \qquad \forall t \\
& x_{ts} \text{ integer}, \;\; y_{ts} \in \{0, 1\} \\
& x_{ts}, y_{ts} \;\; \text{Non-anticipative}
\end{aligned}
$$

"Pretty much" the same as lot sizing model, except ....that production quantities are in increments of *b*

Illustrative Computations

Table 1:

Prob	**B&P time**	**B&P nodes**	**CPLEX Time**	**CPLEX Nodes**
16a	1.13	0	1.80	1722
16b	1.15	0	0.71	569
16c	11.6	8	1.8	1626
16d	16.3	11	6.3	5585
16e	13.3	8	0.9	761

These are 5 stage problems

Prob	B&P time	B&P nodes	CPLEX Time	CPLEX Nodes
32a	156	0	>T	$>10^6$
32b	2945	0	>T	$>10^6$
32c	91	8	1110	$>10^6$
32d	1064	11	>T	$>10^6$
32e	403	8	2800	$>10^6$

These are 6 stage problems

7 stage problems with 64 scenarios also

Conclusions

I should reiterate that

- Convexity and Decomposition remain criti-cal
- Special structure, inexact solves, warm starts etc. remain critical.
- Sampling is new to SIP, but will emerge as we solve larger problems

Important Trends which should continue ...

- Algorithmic approach to tree generation and output analysis

- Computer implementations should find easier interfaces with simulation/validation software

For SP, algorithms if there is one word that deserves its own slide it is ...

Scalability

Scalability

Scalability

Scalability

Scalability

And finally,

Two Stage and Multi-stage
Stochastic Integer Programming Problems
Remain One of the
Grand Challenges in Optimization....

Thank you for your interest.

Comments and Questions, Most Welcome!

In appreciation of the SP community ...

Top 5 reasons to work on

Stochastic Programming Problems

5. Can work with "cosmic distances" with-out leaving home!

4. One begins to easily distinguish musicians from mathematicians: one composes; the other "decomposes"

3. One learns that “Log-concavity” has nothing in common with either lumber or cavities!

2. One also learns that “clairvoyance” requires connections in very high places!

1. The word “non-anticipativity” makes you appreciate what President Bush must go through!