Random Thoughts about Optimization

Rong, Ying

12th October 2006

Outline

A Joke

Given a hard question, you can solve it, if and only if

- either you are a very smart person
- or you have one super computer
- or you have lots of computers

Is this a sufficient condition?

The answer is absolute 'NO'.
The counter-example is: you can find smarter people to help you

Venture Capital's questions and your answers

VC's Questions	Your Answers
What is your target customer?	All the people in the world
How much profit eventually?	$1 Billion
How much percentage of shares are you willing to sell	51%
How much money for each share	$00.1 per share

Smart people's questions and your answers

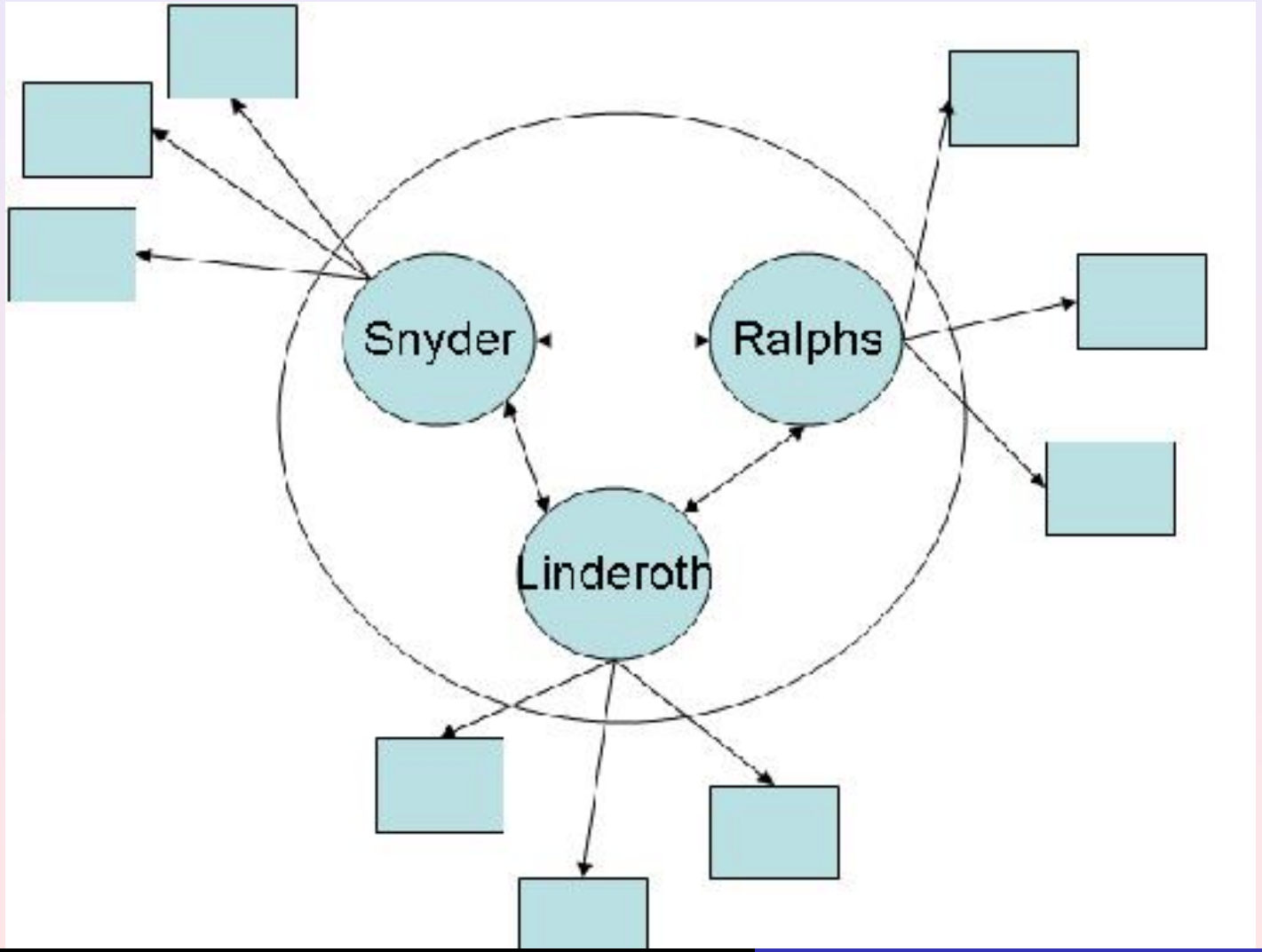
Smart People's Questions	Your Answers
Which journal you aim at?	Science or Nature
How many citations your paper will bring?	1000
What's the sequence of authors?	You can be the first author
How much time I need to put	small amount of time needed

So, smart people are equivalent to venture capital?

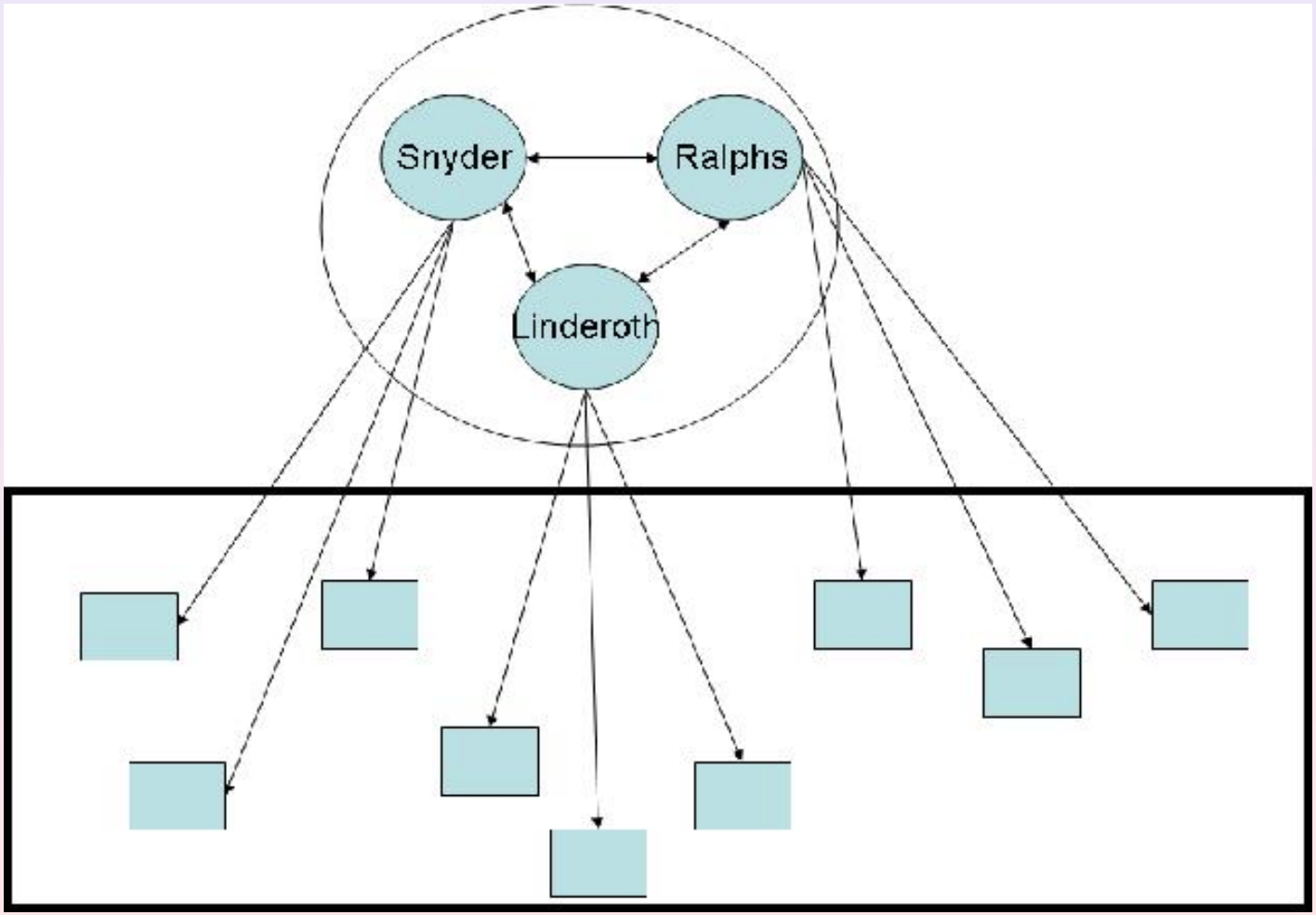
- They allocate their resource on your ideas
- The difference might be smart people do not care your previous answers and they want to help you only because they are inspired by your ideas

Today, I come here to sell my ideas

Current People Grid



Future People Grid



Any relationship between LP and regression

- Danzig: Iteratively reweighted least square
- L1 Estimation

An idea from one of the problems I met in this summer

$$O = D + \alpha(IL - a) + \beta(IP - IL - b) + \gamma S + \epsilon$$

$$s.t. -1 \leq \alpha, \beta \leq 1$$

How to estimate α, β and γ? Let's use L1 estimation

$$\begin{array}{ll} \min & \sum_i |O_i - (c + D_i + \alpha IL_i + \beta(IP_i - IL_i) + \gamma S_i)| \\ s.t. & -1 \leq \alpha, \beta \leq 1 \end{array}$$

How to get the value of estimation

That's a LP problem

$$\min \quad 1^T x$$

$$s.t. \quad \begin{bmatrix} I & I & IL & IP - IL & S \\ I & -I & -IL & -(IP - IL) & -S \end{bmatrix} \begin{bmatrix} x \\ c \\ \alpha \\ \beta \\ \gamma \end{bmatrix} \geq \begin{bmatrix} O - D \\ D - O \end{bmatrix}$$

$$-1 \leq \alpha, \beta \leq 1$$

What's more important?

Hypothesis Test, such as

- $\alpha = \beta$
- $\gamma = 0$

But how can you get the p-value for those hypotheses when you have no idea about the distribution of error (ϵ) information?

If I have the super computer or Grid

1. Bootstrap
2. Solve LP
3. Go back to step 1 for millions of times

If I have smart friends

1. Bootstrap
2. Solve LP
3. One way
    1. Random fix + Random pick
    2. Utilize Local Sensitivity of LP
    3. Go Back to Step 3.1

Need to show what kind of random fix and random pick still give me the same distribution.

If I have smart friends

1. Bootstrap
2. Solve LP to get $(x^*, c^*, \alpha^*, \beta^*, \gamma^*)^T$
3. Another way to use information of LP
   1. Bootstrap
   2. Check whether $(x, c^*, \alpha^*, \beta^*, \gamma^*)^T$ is an extreme point
   3. Check whether it is optimal
   4. Update frequency
4. Go back to step 1

Need to know how to check whether a given point is an extreme point. That's easy! Only check how many constraints are binding.
But how to check whether an extreme point are optimal?
And also how to check a non extreme point are optimal?

What's else?

- what if we want our predict value be bounded?

An idea from Coin-OR workshop

From the talk 'Automated Tuning of Solver Parameters' by Brady Hunsaker:

> *We explore methods for automated tuning of parameters for software solvers. A user specifies a test set of instances and a set of parameters. Our implementation attempts to identify good settings for the parameters without exhaustively trying every combination, using ideas from software testing and machine learning.*

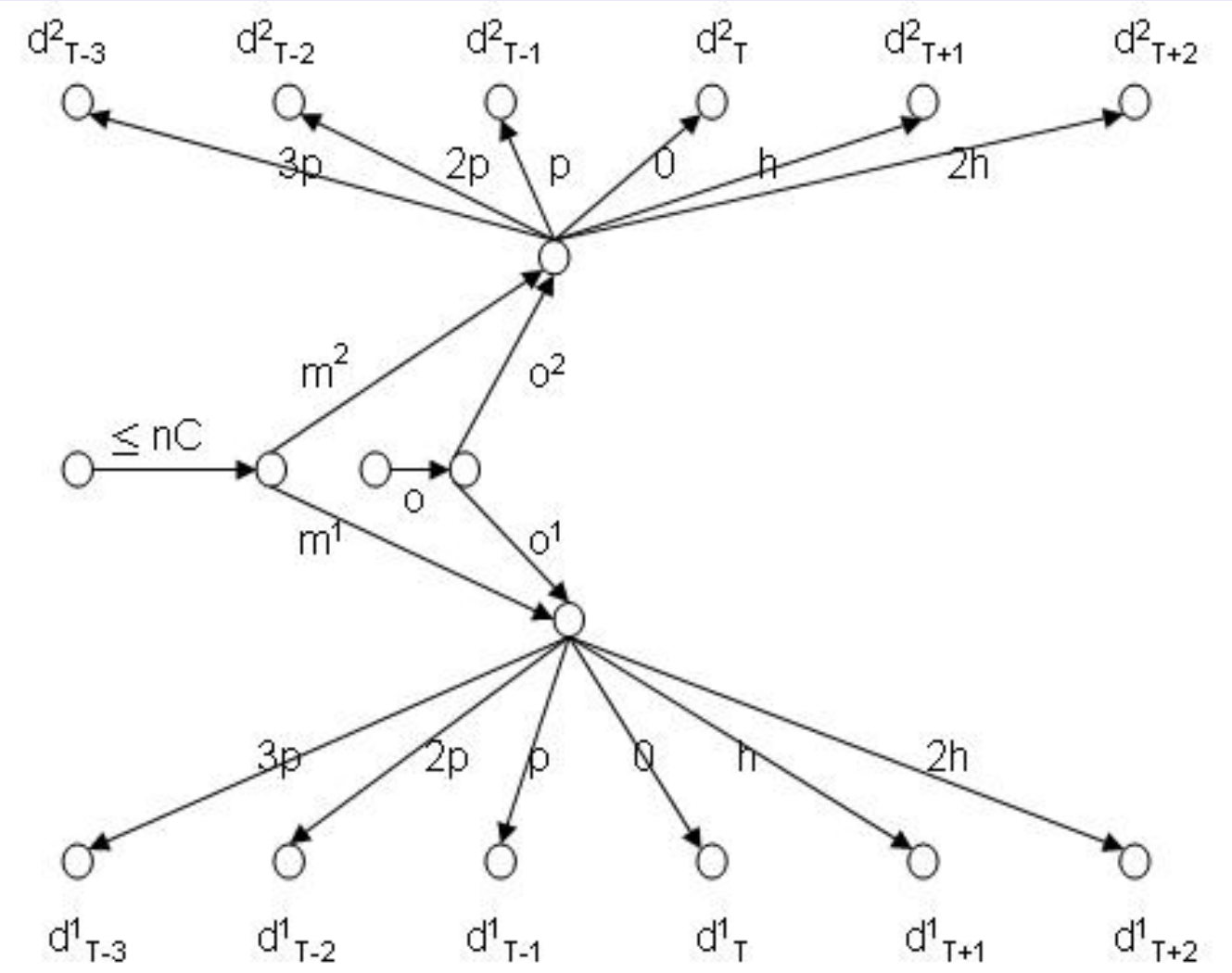
What Brady Hunsaker suggests is the fix value for parameters

Suppose if I know the distribution of the parameters in my model, can I dynamically change the value of parameters based on realized inputs?

An Idea from Agere Project

$$
\begin{array}{lrcl}
\min & \sum_{i,t}(hI^{+}_{i,t} + pI^{-}_{i,t}) + \sum_{k,t} f_k y_{k,t} & + & \sum_{k,t} v_k(\sum_{j=0}^{t} y_{k,j}) \\
 & \sum_{k,i,t} m_{i,k} p_{k,i,t} & + & \sum_{i,t} o_i q_{i,t} \\
 & & & \\
s.t. & I_{i,t-1} + \sum_{k \in K(i)} p_{k,i,t} + q_{i,t} & = & I_{i,t} + d_{i,t} \\
 & \sum_{i \in I(k)} p_{k,i,t} & \leq & c_k \sum_{j=0}^{t} y_{k,t} \\
 & p_{k,i,t}, q_{i,t} & \geq & 0 \\
 & y_{k,t} & \in & N \cup \{0\} \\
 & y_{k,0} & & given
\end{array}
$$

The network formulation



What do I hope?

- To show the equivalence between the network formulation and LP problem
- I hope this formulation may solve LP relaxation faster than Simplex method
- Suppose we change the value of capacity. Is there a quick way to get the new solution as long as the demand is fixed
- Will network formulation give us valuable information about which variable we need to branch
- Will network formulation give us more powerful cuts

iid distribution

Suppose order up level is 10 and capacity is 10 also. Holding cost is 1 and shortage cost is 5
Let's Look at three sequences of demand

1. 15,5,15,5,15,5,15,5
2. 15,15,15,15,5,5,5,5
3. 5,5,5,5,15,15,15,15

The first one has cost $5 * 5 * 4 = 100$. The second one has cost $(5 + 10 + 15 + 20) * 5 = 250$. The third one is $5 + 5 + 5 + 5 + (5 + 10 + 15 + 20) * 5 = 270$

How to do it efficient

- Apply the sampling technique to one dimension demand for 8 times, such as stratified sampling.
- Apply Latin hypercube sampling to 8 dimension demand

Is anything between?
How about we apply the sampling technique to one dimension demand, and then think out a way to do the partial permutation?

Markov Process

Let's imagine the following Markov process.

	1	-1
1	0.4	0.6
-1	0.6	0.4

Suppose you need to do the finance decision based this underlying process real time. And you have to simulate different sample path to get your decision at the current periods. How are you going to do to reduce the variance?

We cannot blindly use Latin hypercube sampling, since we don't have square for the sample space.

COR@L eggs pool