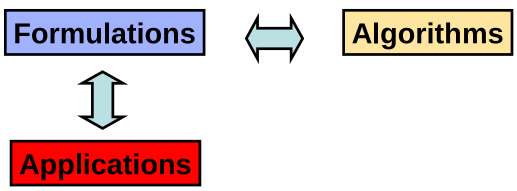
Optimization: Formulations, Algorithms and Applications

Vassilis S. Vassiliadis©
6 & 13 May 2011
Department of Chemical Engineering and Biotechnology
University of Cambridge

"Optimization is the science of finding the best solution"

Roger Fletcher,
Practical Methods of Optimization,
John Wiley & Sons, (2000).

Main areas of presentation



Overview

1. Optimization problem statement
2. Convexity and nonlinearity
3. Variable types
4. The "curse of dimensionality"
5. Generalised optimization problems
6. Important modern algorithms
7. Derivative-free optimization
8. Conclusions

"Make things as simple as possible, but not simpler."

Albert Einstein, 1930's

1. Optimization problem statement

• Objective function: $\min_x \text{ or } \max_x f(x)$	• Inequality constraints $g(x) \le \text{ or } \ge 0$
• Equality constraints $h(x) = 0$	• Simple bounds $x^L \le x \le x^U$

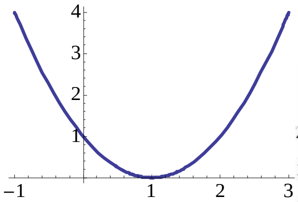
- General Mathematical Programming Problem (MP)
- If all variables are continuous,
  Nonlinear Programming Problem (NLP)

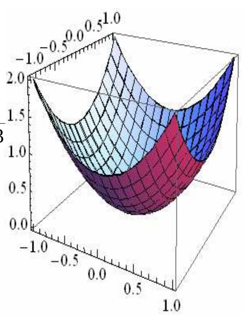
- Objective: usually an economic criterion
  - Minimum cost
  - Maximum profit, revenue
- Equality constraints
  - Modelling equations: LAE’s, NLAE’s, ODE’s, PDE’s
- Inequality constraints
  - Operating limitations
  - Quality control
- Bounds
  - Implied by inequality constraints
  - Usually arise naturally from the problem definition

2. Convexity and nonlinearity

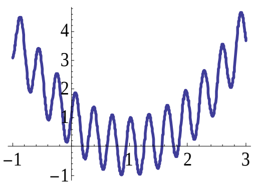
- ***Nonlinearity***, wherever it arises
  - Complicates solution
  - But in itself is not the most important complicating factor
- ***Nonconvexity*** of the objective and/or of the constraint set
  - Serious complication in practical applications

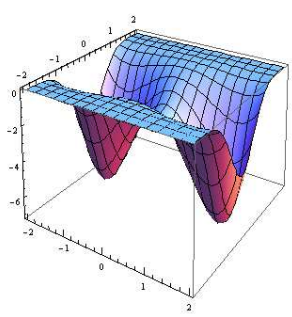
- Convex functions



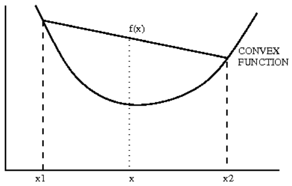


- Nonconvex functions

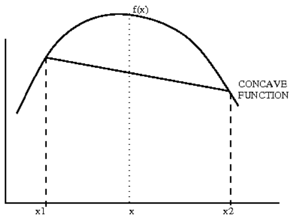




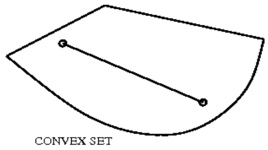
- A convex function is always overestimated by its chord
  - For minimization it is a guarantee of solution uniqueness



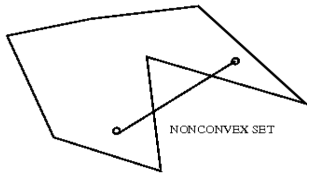
- A concave function is underestimated by its chord
  - For maximimization it is a guarantee of solution uniqueness



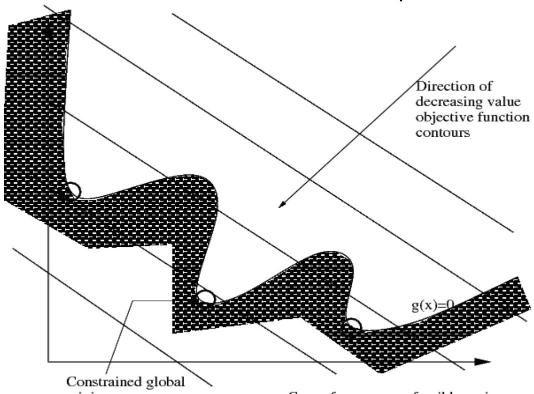
- For all optimization problems we want the constraints to define a convex set



  - A convex set contains all the points of the line connecting two of its points
  - *i.e.* it contains weighted averages of points!



- Even if the objective is convex (for min), if the constraints are nonconvex → multiple minima



Case of nonconvex feasible region

- So what can we solve?
- Many things, even non-differentiable problems, but…
- General NLP will have no certificate of global optimality
- Special algorithms for general problems exist
  - Certificate of global optimality comes at high cost
    - Long computation times, smaller problems
- Convex Programming Problem
  - The only one that has a certificate of global optimality

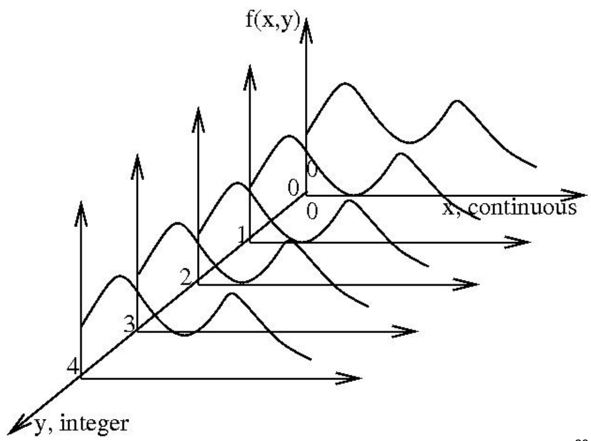
- Convex Programming Problem:

$$\min_x \quad f(x) \;\; \text{convex objective}$$
$$\textit{subject to:}$$
$$h(x) = 0 \;\; \text{affine equalities} \;\Rightarrow\; Ax = b$$
$$g(x) \le 0 \;\; \text{convex inequalities}$$
$$x \quad \text{continuous variables}$$

- Only one solution: the global one
- Solved in polynomial time (1995 proof)

3. Variable Types

- Typical case is where variables belong to the real set of ***continuous*** numbers, R^N
- In many applications of engineering interest the variables are ***integers***
  - A special case is ***binary*** variables,
    - used to encapsulate boolean operations (absence of presence of a unit, on/off operations, *etc*.)
- Variables (arguments) are ***functions*** -- later

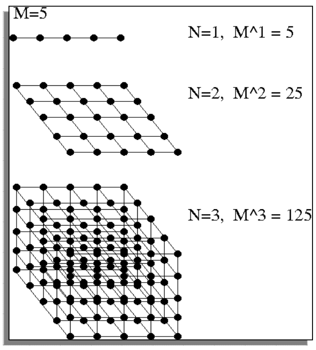


4. The “curse of dimensionality”

“It is a scale of proportions which makes the bad difficult and the good easy .”

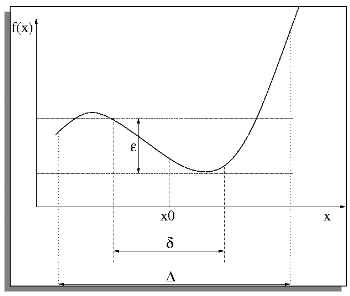
Albert Einstein, 1946

- If we assign 10 intervals for each variable
- With 100 variables
- Evaluating once the function in each compartment means
- 10^{100} function evaluations
- A bad algorithm, runs in exponential time

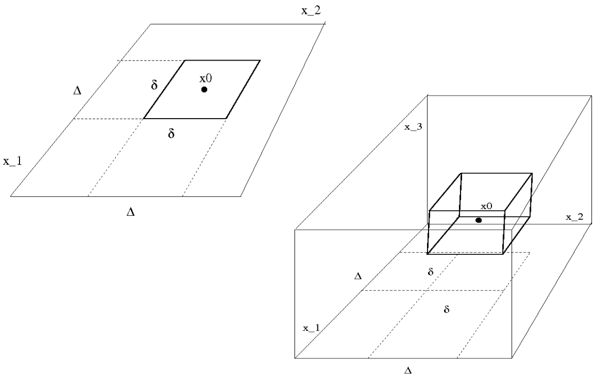


4.1 Information density ‘dilution’ with dimensionality

- Function *f(x)*
- In 1-D, interval of interest Δ
- A single point, characterises an interval $\delta < \Delta$
- with maximum deviation ε in the function value



- If the same values hold for many dimensions then



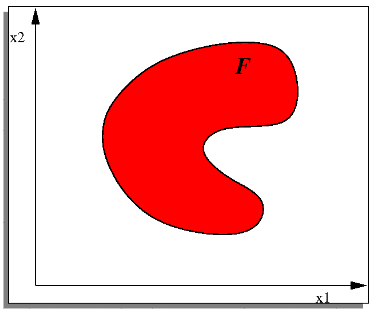
- Fraction of hyper-volume characterised by a single function evaluation is

$$r = \left(\frac{\delta}{\Delta}\right)^N$$

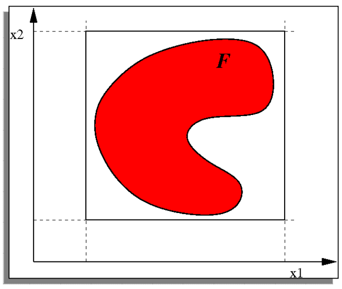
- As, $0 < \frac{\delta}{\Delta} < 1$
- Then the fraction decays exponentially with the number of dimensions
- Or, conversely, we need an exponential number of evaluations to characterise a percentage of the search volume

4.2 The probability of satisfying constraints with random sampling

- 2-D feasible region, for example



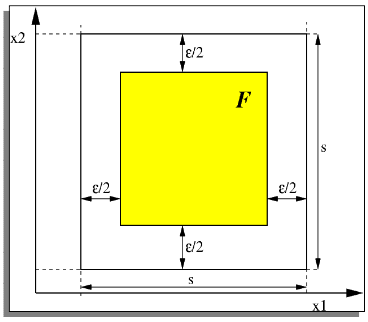
- difficult with random sampling methods to get a point within arbitrary feasible regions
- find an outer bounding box containing the region
- sample uniformly points from within it



1. If the point is feasible,
2. And if it improves the objective
3. We keep it,
4. Else, we pick a new random point,
   - Go To step 1

- Will this algorithm work for any dimensions?

- For simplicity, consider a square feasible region, of side (*s*-*ε*)
- bounded within an outer square sampling box, of side *s*

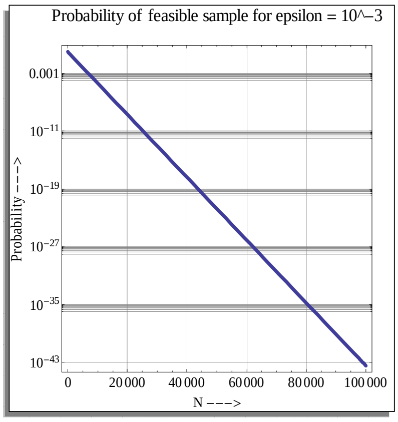


- The probability of finding a point within the inner square, is equal to the ratio of the two hyper-volumes:

$$P(x \in F) = \frac{(s-\varepsilon)^N}{s^N} = \left(1 - \frac{\varepsilon}{s}\right)^N$$

- Which again decays exponentially,
  - even if $\varepsilon/s \ll 1$
- Higher dimensional objects have all of their volume near the surface!

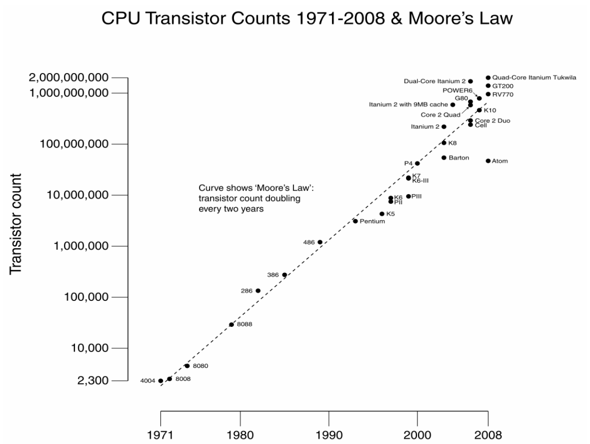
- For $\varepsilon/s = 10^{-3}$ we obtain the following figure
- For example, for N=10,000 P=0.000045
- Finding a *single feasible point* requires an exponential number of trials!



- Generating points within a convex feasible set is generally not a problem
- Easy if problem is defined by convex inequalities
- Randomized methods can work
- Generating points in nonconvex sets can be an issue

4.3 Faster computers?

- Will faster computers be the answer to addressing the problems of high dimensionality?
- Computer chips double speed (transistors) every 2 years (Moore's Law)



From Wikipedia

- Even if Moore's law does not saturate (predicted by 2020),
- Doubling the speed of a computer every 2 years,
- For an exponential complexity of $O(2^N)$ operations for a problem of N variables,
- This means we will be able to solve a problem of N+1 variables
  - In the same time we solved the smaller problem, 2 years before!
- As we will see, there are fortunately special algorithms to deal with combinatorial problems

5. Generalised optimization formulations

"Most of the fundamental ideas of science are essentially simple, and may, as a rule, be expressed in a language comprehensible to everyone ."

Albert Einstein

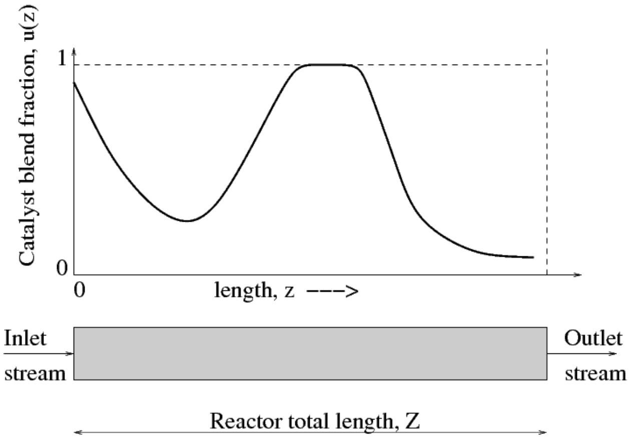
- We shall consider next some generalised formulations
- The NLP we introduced on the first slide sets the general format
- The scope of optimization models can be widened to capture problems of significant interest in industry and research

5.1 Optimal Control Problems (OCP)

- The case where we are looking for control functions
- State responses are also functions
- → *infinite dimensional optimization problems*
- Best shown first by example

5.1.1 Bifunctional catalyst PFR

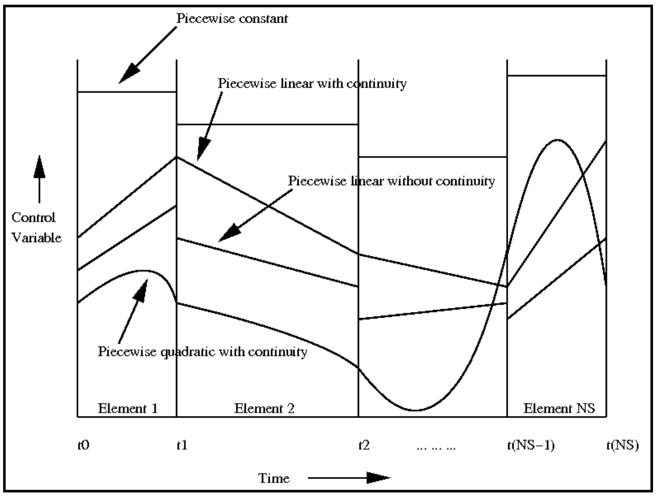
- A catalytic tubular reactor design problem, in hydrocarbon processing
- using a bi-functional catalyst blend along its length,
- the task is to find the optimal catalyst blend,
- to maximise the yield of a desired product at the outlet of the reactor

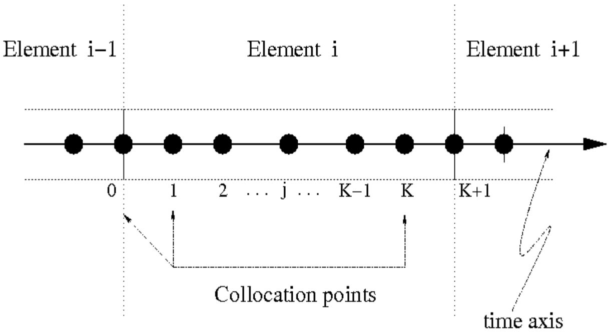


5.1.2 Parameterization methods

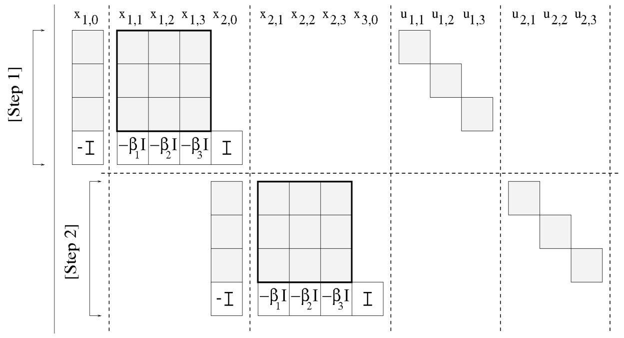
- Use of parameterization is made so as to make the problems finite dimensional NLP's
- Either parameterize control functions only,
  - Control vectror parameterization method (CVP)
- Or parameterize controls and state variables simultaneously
  - Simultaneous approach, using collocation over finite elements

43



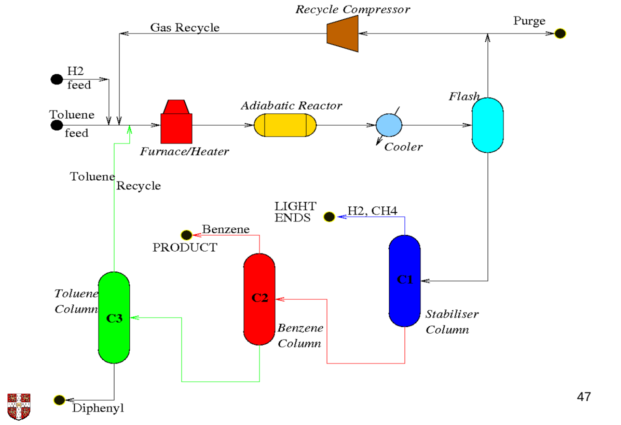


45

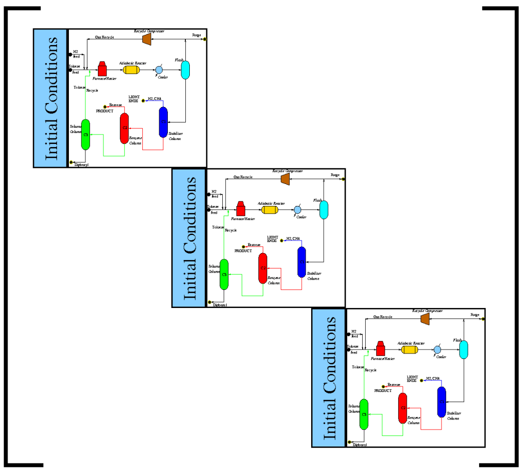


46

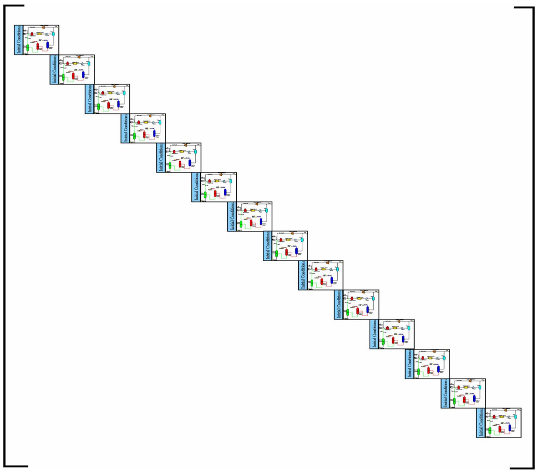
- Example: given a dynamic flow sheet:



47



48



49

5.1.3 OCP formulation

$x(t)$ Differential state variables

$y(t)$ Algebraic state variables

$u(t)$ Control variables

p Time-invariant design parameters

t_f Final time

50

$$\min_{x(t),y(t),u(t),p,t_f} \phi(x(t_f), y(t_f), u(t_f), p, t_f)$$

$$subject\ \ to:$$

$$\frac{dx}{dt}(t) = f(x(t), y(t), u(t), p, t); \quad t_0 \le t \le t_f$$

$$h(x(t), y(t), u(t), p, t) = 0; \quad t_0 \le t \le t_f$$

$$x(t_0) = x(t_0, p)$$

51

$$g(x(t), y(t), u(t), p, t) \le 0; \quad t_0 \le t \le t_f$$

$$u^L \le u(t) \le u^U; \quad t_0 \le t \le t_f$$

$$p^L \le p \le p^U$$

$$t_f^L \le t_f \le t_f^L$$

52

5.2 Multiobjective optimization (MOO)

- Multiobjective optimization deals with problems with at least 2 objectives
- Multiple objectives have to be optimized *simultaneously*
- They are conflicting performance indices (targets) for a given design
  - Maximize profit,
  - minimize environmental impact,
  - minimize risk, *etc*.

53

$$\min_x \quad f = \left(f_1(x), f_2(x), \dots, f_m(x)\right)^T$$

$$subject\ to:$$

$$h(x) = 0$$

$$g(x) \le 0$$

54

- The value of the objective vector f^* is termed *Pareto optimal* when
- There ***does not exist*** another vector *f* such that

$$f_i \leq f_i^* \quad \text{for all} \quad i \in \{1,2,\ldots,m\}$$

and

$$f_j < f_j^* \quad \text{for at least one} \quad j \in \{1,2,\ldots,m\}$$

- *i.e.* does not improve at least one objective while the others remain at least the same
  - Solution cannot be '*dominated*'

5.2.1 MOO Solution methods

- There are a number of methods to solve MOO problems.
- Here we will present only an intuitive approach that converts an MOO problem into an NLP problem – scalarization
- Aggregate objective function (AOF)
  - Assign weights to different objectives
  - Combine them into a cumulative one
  - Subjective method as weights are based on experience of relative importance of objectives

$$\min_x \quad \phi(x) = \sum_{i=1}^{m} \omega_i f_i(x)$$

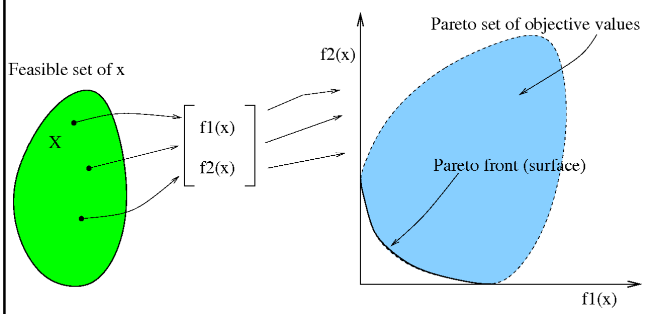
subject to:

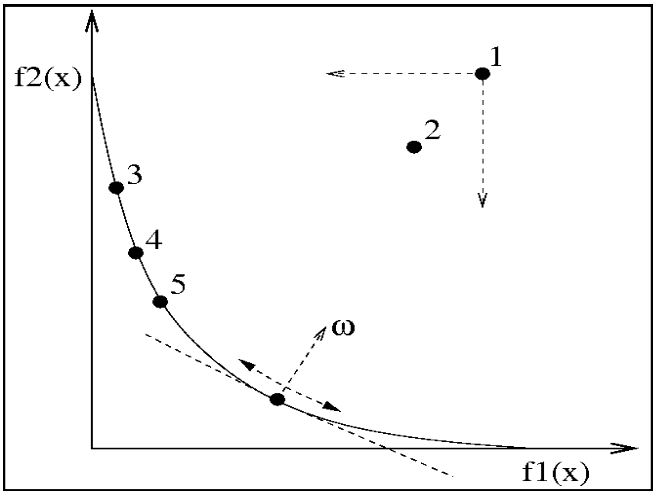
$$h(x) = 0$$

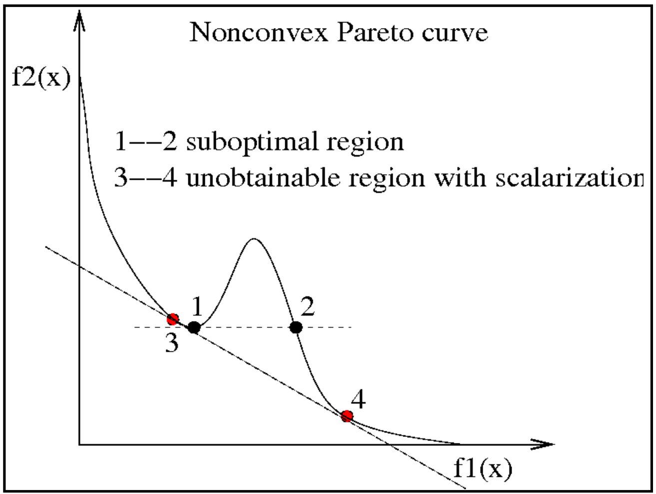
$$g(x) \leq 0$$

such that :

$$\sum_{i=1}^{m} \omega_i = 1; \quad 0 \leq \omega_i \leq 1$$







5.3 Bilevel optimization (BLP)

- A bilevel programming problem
  - Contains a subset of the variables, which is required to be an optimal solution of a second optimisation problem.
- Best demonstrated first by an application

61

5.3.1 Predicting Genetic Modificatons

- Modification of biochemical reaction (metabolic) pathways
- To achieve an improved productivity of a given metabolite
- Using model-based techniques to propose alternative genes for knock-out
- Save experimental costs and time

62

- If we had kinetics for ***all the reactions in metabolic pathways***, then
  - Measure sensitivity of process output to given enzyme concentrations
  - Knock-out genes that code for metabolites that
    - Inhibit reactions on productive pathway
    - Consume metabolic products needed in productive pathway
  - For accuracy, we would also need to know genome control mechanisms (operons) on metabolic reactions

63

Such detailed kinetic information is not available

- As such, although can be viewed as ordinary chemical reaction pathways,
- Special handling is needed to predict reliable modifications

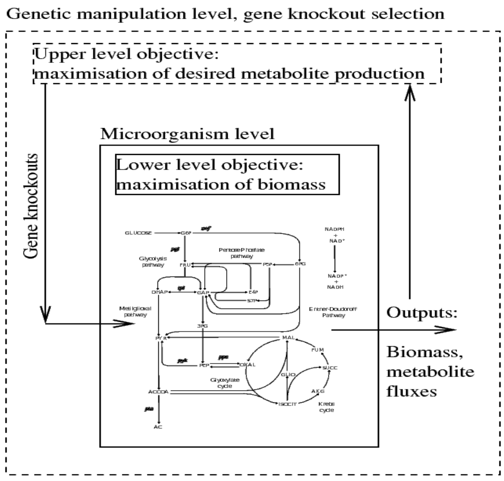
64

- Idea:
  - Use fluxes through metabolic pathways
  - Use stoichiometry of reactions involved (known)
  - Find native state of fluxes
  - Predict redistribution of fluxes subject to a gene knock-out.
    - Redistribution prediction is the key here
    - Fluxes would redistribute according to which criterion?

Reference:

Burgard, A.P., Pharkya, P., Maranas, C.D., "OptKnock: A Bilevel Programming Framework for Identifying Gene Knockout Strategies for Microbial Strain Optimization", Biotechnology and Bioengineering, 84(6), 647–657, (2003).

65



66

- Thus a BLP is an optimization problem which has
  - as one of the constraints being itself an optimisation problem (nested problem)
  - We can have multilevel optimization problems (generalisation)
  - BLP's are solved by suitable transformation into NLP's

67

5.3.2 BLP Formulation

$$\begin{aligned}
&\min_{x,y} \; F(x,y) \\
&\textit{subject to}: \\
&\quad H(x,y) = 0 \\
&\quad G(x,y) \le 0 \\
&\quad \min_{y} \; f(x,y) \\
&\quad \textit{subject to}: \\
&\quad\quad h(x,y) = 0 \\
&\quad\quad g(x,y) \le 0
\end{aligned}$$

68

"God does not play dice with the Universe."

"The more success the quantum theory has, the sillier it looks."

Albert Einstein

5.4 Stochastic Optimization (Programming) (SP)

Optimization under uncertainty

- 3 sources of uncertainty in Chemical Engineering processes:

1. Input variability,
2. Disturbances,
3. Parametric uncertainty.

70

- Results in designs that select decision (design) variables so as to
  - Optimize expectation of the objective index
  - Satisfy exactly hard constraints
  - Soften up constraints of qualitative nature into
    - probability of satisfaction
    - Satisfaction of their average value
- The above ingredients may be all present or in part in resulting formulations

71

5.4.1 SP Formulations

x Decision variables, $\in X \subset R^{n_x}$

ξ Uncertain parameters (random variables), $\in \Xi \subset R^{n_\xi}$ distributed according to a (joint) PDF, $P(\xi)$

- The PDF may be continuous or discrete, accordingly also the set Ξ

72

- Main formulation:

$$\min_x \bar{f}(x) = E[f(x,\xi)] = \int_{\Xi} f(x,\xi)\, dP(\xi)$$

$$subject\ to:$$

$$h(x,\xi) = 0$$

$$g(x,\xi) \le 0$$

73

- Focusing on feasibility of the inequality constraints (regarding model equalities as hard constraints here)
  - A choice of decision variables may not satisfy them for all realisations of the uncertain parameters
  - We need to define a 'looser' type of feasibility

74

- 1st relaxation: average value of constraints

$$\bar{g}(x) = E[g(x,\xi)] \le 0$$

- 2nd relaxation: probability of satisfaction

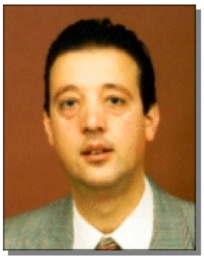
$$P\big(g(x,\xi) \le 0\big) \ge \varepsilon, \quad 0 < \varepsilon \le 1$$

75

- Solution of general SP problems is often based on
  - Discrete samples
  - Quadrature forms for integral evaluations
  - Multiple scenarios realisations
  - Resulting in standard NLP formulations

76

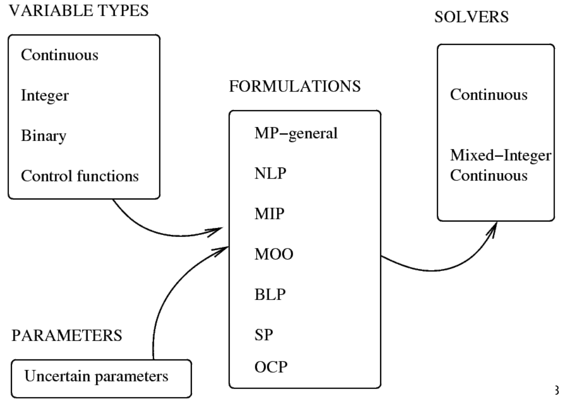
5.4.2 SP people…



E.N. (Stratos) Pistikopoulos
CPSE Director, Imperial College London
Professor of Chemical Engineering,
Department of Chemical Engineering
Imperial College London

Specialization
Optimization under uncertainty

Summary of formulations



3

6. Important modern algorithms

79

"To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science ."

Albert Einstein

- There are 3 types of algorithms associated with all applications shown thus far, and with others that will be highlighted in this section:
  1. Solution of large LP/NLP problems
  2. Solution of Mixed-Integer problems
  3. Deterministic Global Optimization

81

6.1 Interior Point Methods (IPM's) for LP and NLP problems

- The most important problems first addressed by optimization methods were LP's
- LP's
  1. Have a convex feasible region defined by linear constraints and bounds (if the constraints are not conflicting)
  2. The optimal point occurs at an extreme point of the feasible region, *i.e.* a vertex
  3. There is only a single, global optimum

82

- Let us consider a general LP problem formulation to begin with:

$$\min_x z = c^T x = \sum_{i=1}^{n} c_i x_i$$

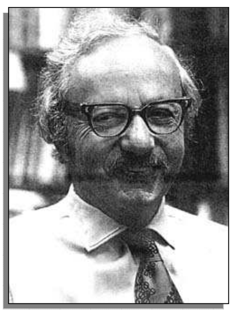
$$subject\ to:$$

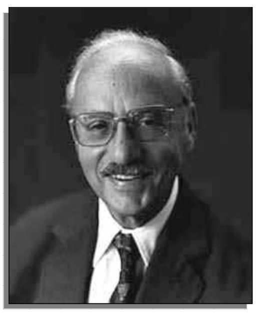
$$a_i x \geq b_i,\ i = 1,2,\ldots,m$$

$$x_i \geq 0,\ i = 1,2,\ldots,n$$

83

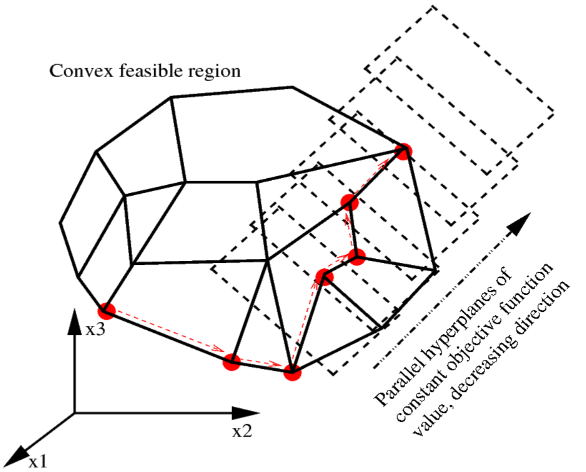
- George Bernard Dantzig (November 8, 1914 – May 13, 2005): The simplex algorithm for LP

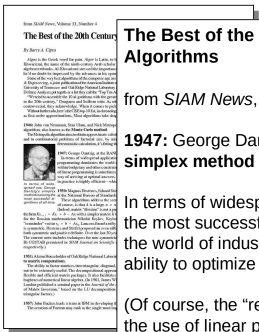




84

- Idea (simplex algorithm):
  1. Find a way to identify vertices (elimination operations among the constraints)
  2. Evaluate the objective function value
  3. Find a way to identify *adjacent vertices* from current one
  4. Evaluate a measure of how much each of these adjacent vertices would improve the objective function value
  5. Move to one that improves the objective
  6. Go To 2, until no improving vertices can be found → global solution found





The Best of the 20th Century: Editors Name Top 10 Algorithms

from *SIAM News*, Volume 33, Number 4

1947: George Dantzig, at the RAND Corporation, creates the **simplex method for linear programming**.

In terms of widespread application, Dantzig's algorithm is one of the most successful of all time: Linear programming dominates the world of industry, where economic survival depends on the ability to optimize within budgetary and other constraints.

(Of course, the "real" problems of industry are often nonlinear; the use of linear programming is sometimes dictated by the computational budget.)

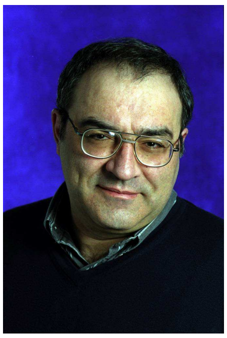
The simplex method is an elegant way of arriving at optimal answers. Although theoretically susceptible to exponential delays, the algorithm in practice is highly efficient—which in itself says something interesting about the nature of computation.

- The question was: are there any polynomial time algorithms for LP
- Worst case scenario for simplex is exponential time -- *e.g.* $O(2^N)$
- Some methods were found that would operate in polynomial time, $O(N^k)$, but k was large
- This was until 1979 and then 1984…

The polynomial time revolution

Leonid Genrikhovich Khachiyan
(May 3, 1952 – April 29, 2005)

- The ellipsoid algorithm



Rutgers The State University of New Jersey

News

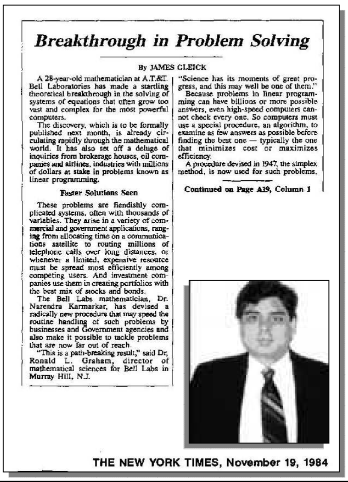
World Renowned Computer Scientist Leonid G. Khachiyan Dies at 52

May 03, 2005

Khachiyan proved the existence of an efficient way to solve linear programming problems thought to be intractable until that time.

His 1979 breakthrough dealt with the underlying mathematics, opening doors beyond linear programming to what is known as combinatorial optimization – finding the best of a finite, but often astronomically large, number of options.

- Narendra Karmarkar, "A New Polynomial Time Algorithm for Linear Programming", Combinatorica, 4(4), 373–395, (1984).
- "affine scaling algorithm"



Breakthrough in Problem Solving

By JAMES GLEICK

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

Faster Solutions Seen

These problems are fiendishly complicated systems, often with thousands of variables. They arise in a variety of commercial and government applications, ranging from allocating time on a communications satellite to routing millions of telephone calls over long distances, or whenever a limited, expensive resource must be spread most efficiently among competing users. And investment companies use them in creating portfolios with the best mix of stocks and bonds.

The Bell Labs mathematician, Dr. Narendra Karmarkar, has devised a radically new procedure that may speed the routine handling of such problems by businesses and Government agencies and also make it possible to tackle problems that are now far out of reach.

"This is a path-breaking result," said Dr. Ronald L. Graham, director of mathematical sciences for Bell Labs in Murray Hill, N.J. "Science has its moments of great progress, and this may well be one of them."

Because problems in linear programming can have billions or more possible answers, even high-speed computers cannot check every one. So computers must use a special procedure, an algorithm, to examine as few answers as possible before finding the best one — typically the one that minimizes cost or maximizes efficiency.

A procedure devised in 1947, the simplex method, is now used for such problems.

Continued on Page A19, Column 1

THE NEW YORK TIMES, November 19, 1984

- Patent followed
- Software & hardware developed
- The algorithm was original and theoretically sound, but there was more to follow…

AT&T Markets Problem Solver, Based On Math Whiz's Find, for $8.9 Million

By ROGER LOWENSTEIN
Staff Reporter of THE WALL STREET JOURNAL

NEW YORK—American Telephone & Telegraph Co. has called its math whiz, Narendra Karmarkar, a latter-day Isaac Newton. Now, it will see if he can make the firm some money.

Four years after AT&T announced an "astonishing" discovery by the Indian-born Mr. Karmarkar, it is marketing an $8.9 million problem solver based on his invention.

Dubbed Korbx, the computer-based system is designed to solve major operational problems of both business and government. AT&T predicts "substantial" sales for the product, but outsiders say the price is high and point out that its commercial viability is unproven.

"At $9 million a system, you're going to have a small number of users," says Thomas Magnanti, an operations-research specialist at Massachusetts Institute of Technology. "But for very large-scale problems, it might make the difference."

Korbx uses a unique algorithm, or step-by-step procedure, invented by Mr. Karmarkar, a 32-year old, an AT&T Bell Laboratories mathematician.

"It's designed to solve extremely difficult or previously unsolvable resource-allocation problems—which can involve hundreds of thousands of variables—such as personnel planning, vendor selection, and equipment scheduling," says Aristides Fronistas, president of an AT&T division created to market Korbx.

Potential customers might include an airline trying to determine how to route many planes between numerous cities and an oil company figuring how to feed different grades of crude oil into various refineries and have the best blend of refined products emerge.

AT&T says that fewer than 10 companies, which it won't name, are already using Korbx. It adds that, because of the price, it is targeting only very large companies—mostly in the Fortune 100.

Korbx "won't have a significant bottom-line impact initially" for AT&T, though it might in the long term, says Charles Nichols, an analyst with Bear, Stearns & Co. "They will have to expose it to users and demonstrate" it uses.

AMR Corp.'s American Airlines says it's considering buying AT&T's system. Like other airlines, the Fort Worth, Texas, carrier has the complex task of scheduling pilots, crews and flight attendants on thousands of flights every month.

Thomas M.Cook, head of operations research at American, says, "Every airline has programs that do this. The question is: Can AT&T do it better and faster? The jury is still out."

The U.S. Air Force says it is considering using the system at the Scott Air Force Base in Illinois.

One reason for the uncertainty is that AT&T has, for reasons of commercial secrecy, deliberately kept the specifics of Mr. Karmarkar's algorithm under wraps.

"I don't know the details of their system," says Eugene Bryan, president of Decision Dynamics Inc., a Portland, Ore., consulting firm that specializes in linear programming, a mathematical technique that employs a series of equations using many variables to find the most efficient way of allocating resources.

Mr. Bryan says, though, that if the Karmarkar system works, it would be extremely useful. "For every dollar you spend on optimization," he says, "you usually get them back many-fold."

AT&T has used the system in-house to help design equipment and routes on its Pacific Basin system, which involves 22 countries. It's also being used to plan AT&T's evolving domestic network, a problem involving some 800,000 variables.

THE WALL STREET JOURNAL, August 15, 1988

- Gill, Philip E.; Murray, Walter, Saunders, Michael A., Tomlin, J. A. and Wright, Margaret H. "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method". Mathematical Programming 36(2): 183–209, (1986).
- A seminal paper in optimization in modern times
- Initiated the interior point / barrier method revolution both for LP and NLP

- Effectively showed that Karmarkar's method was equivalent to employing an old technique for NLP from the 1960's
  - – Use of logarithmic barrier functions to "absorb" inequality constraints into objective function (similar to the use of penalty functions)
- Studied, among many others, by Anthony V. Fiacco and Garth P. McCormick
  - – Nonlinear Programming – Sequential Unconstrained Minimization Techniques Anthony V. Fiacco and Garth P. McCormick Published by Society for Industrial & Applied Mathematics, new edition 1987, originally published 1968.

- Barrier transformation of inequality constrained problem into sequence of unconstrained problems (the following is called a *primal* IPM)

$$\min_x f(x) \quad \text{subject to}: \quad g_i(x) \geq 0,\ i = 1,2,\ldots,m$$

$$\min_x \phi(x,\mu) = f(x) - \mu \sum_{i=1}^{m} \ln\left(g_i(x)\right)$$

- Algorithm
  1. Start from a large value of the barrier parameter μ;
     - problem does not feel influence of objective
  2. Minimize unconstrained barrier objective function
  3. Reduce μ,
     - small values of μ decrease the influence of the constraints, and this only becomes important when close to the boundary
  4. Minimize barrier objective starting from previous minimizer
  5. Go To 3, until sufficiently close to constrained minimum

- Log- barrier methods only managed to solve small problems in the 60's and 70'
  - severe ill-conditioning as μ <<
  - used optimization methods for general unconstrained optimization problems
  - Converged the unconstrained subproblems completely
- Were eventually abandoned
  - new methods emerged for NLP (Sequential Quadratic Programming, SQP),
  - simplex was the only method for LP
- Since 1986, modern Newton methods were used, with advanced Linear Algebra codes employed
- To achieve polynomial time complexity:
  - Each time μ is decreased perform *only 1 Newton step*

- Small LP example

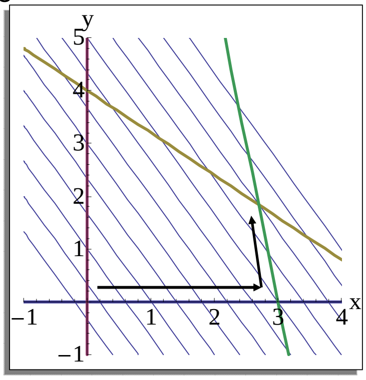
$$\min_{x,y} -5x-3y$$

$$\textit{subject to}:$$

$$4x+5y \le 20$$

$$6x+1y \le 18$$

$$x \ge 0,\ y \ge 0$$



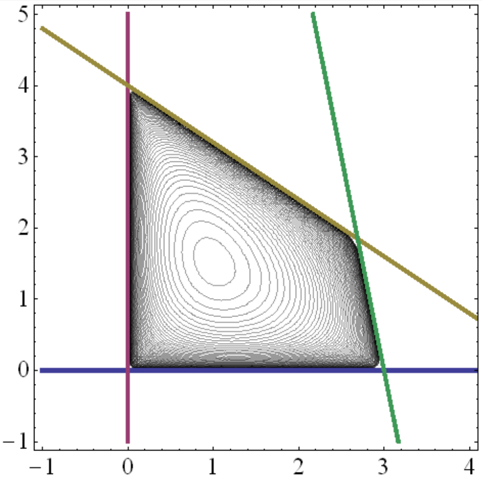
Adopted from: YouTube, **InteriorPointMethodDemonstration.wmv**

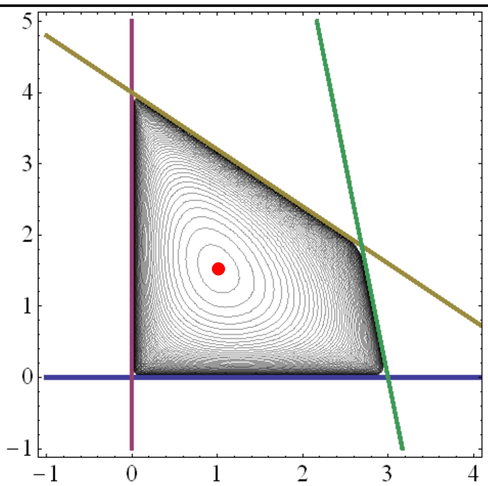
- Transformation into an unconstrained sequence:

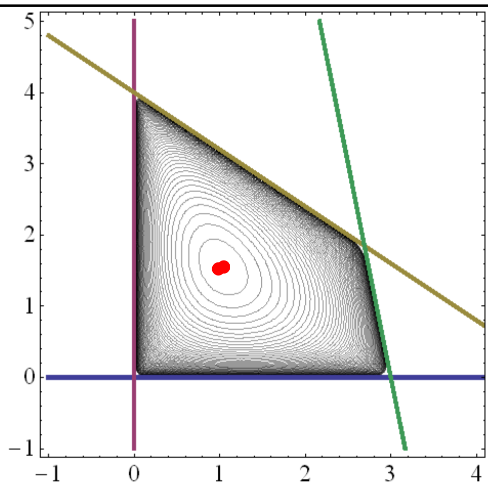
$$\min_{x,y} -5x-3y-\mu\big(\ln(20-4x-5y)+ \ln(18-6x-1y)+ \ln(x)+\ln(y) \ \big)$$

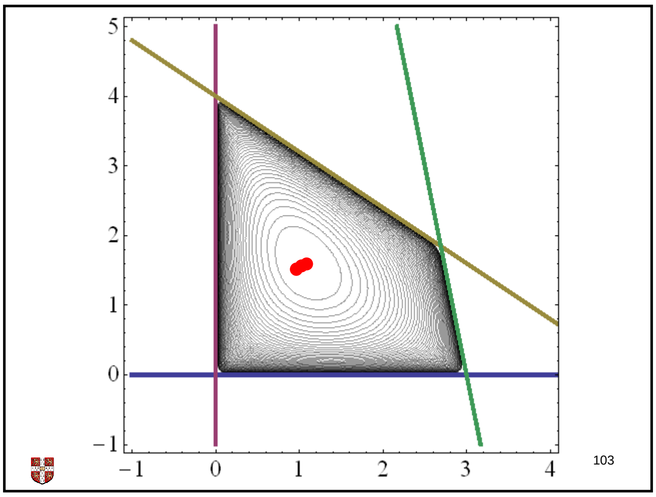
- In the following slides:
  - μ starts at 100.0
  - is reduced by a factor of 2.0
  - for k = 10 iterations
  - Points shown are minimizers (central path) not the path produced by IPM solvers

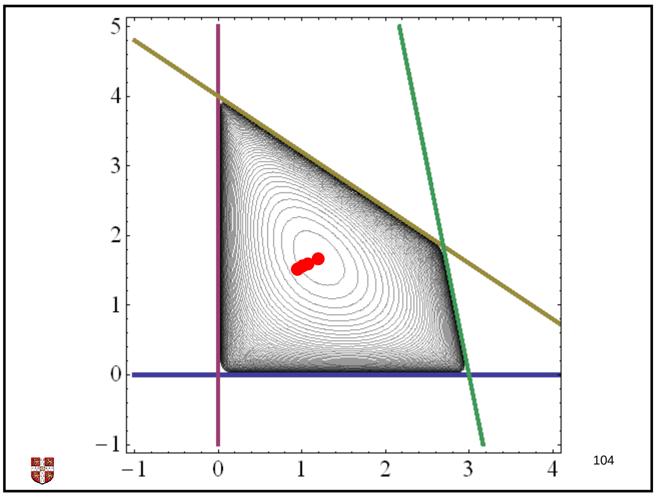
start

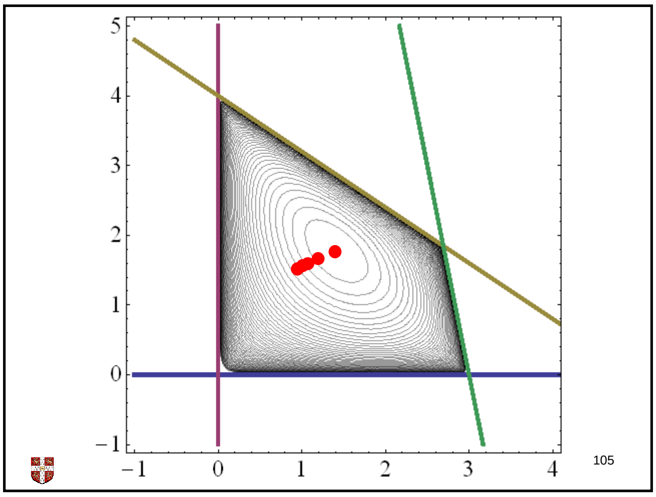


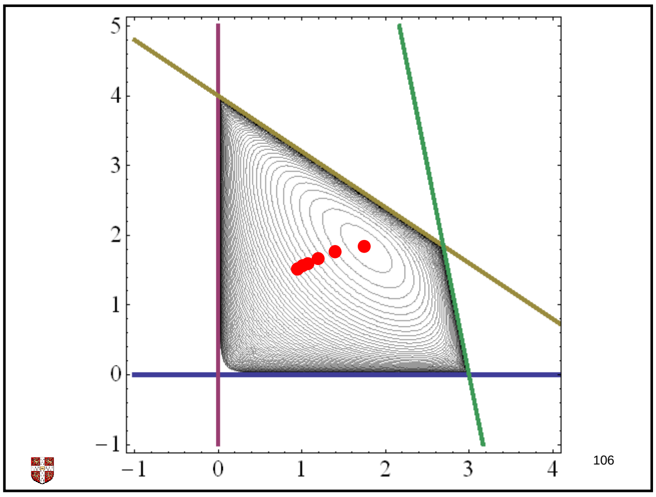


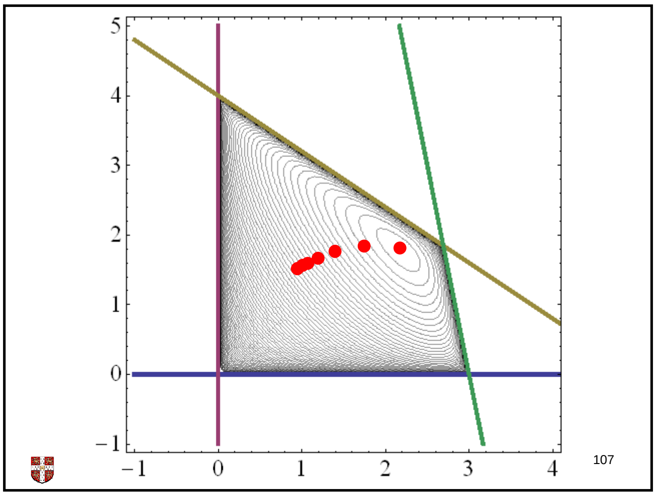


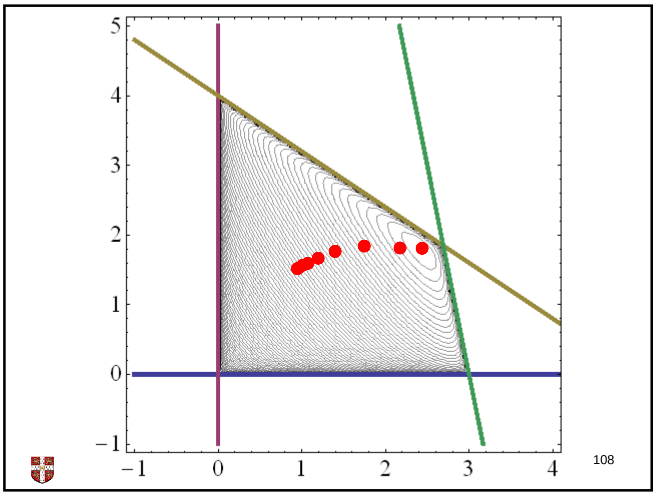

103

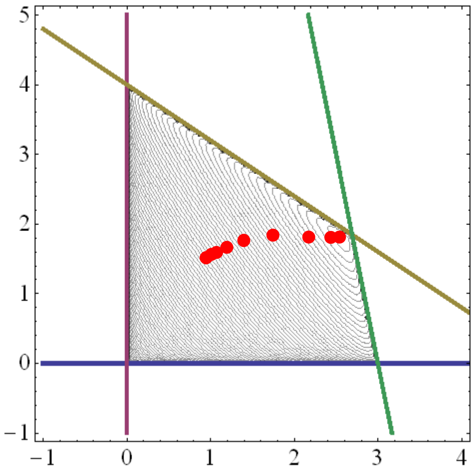

104


105


106

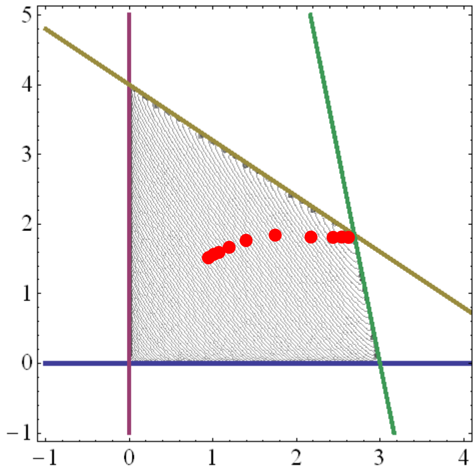

107


108



109

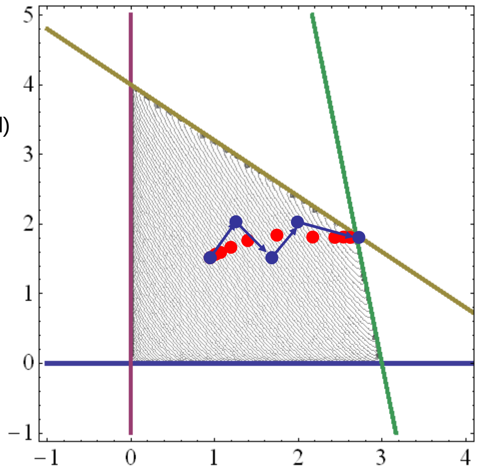
end



110

IPM path

(not actual)



111

- The primal barrier method has some shortcomings
  - Primarily that it needs a feasible point to start
  - Nonlinear inequality constraints cannot be guaranteed to remain feasible during iterations
- Almost exclusively replaced nowadays by the *primal-dual* barrier methods
  - All NLP's (and LP's) can be cast into a "canonical form", involving
  - Equality constraints which can be violated
  - Bounds (always satisfied, easily initialized)

112

- Canonical NLP problem:

$$\min_x f(x)$$
$$subject\ to:$$
$$h(x) = 0$$
$$x \geq 0$$

- Equality constrained problem sequence

$$\min_x \phi(x,\mu) = f(x) - \mu \sum_{i=1}^{n} \ln(x_i)$$
$$subject\ to:$$
$$h(x) = 0$$

113

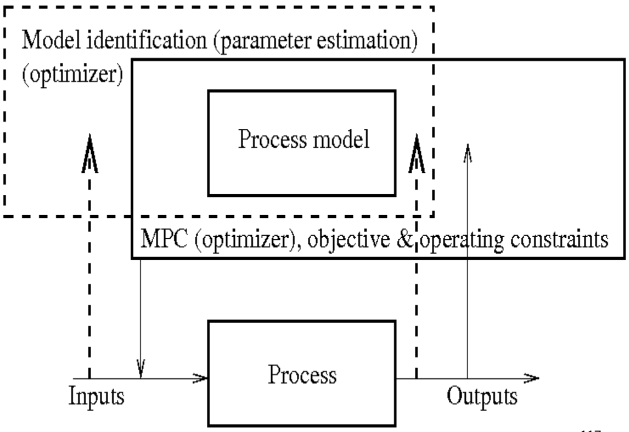
- IPM's: a success story
  - Solve problems of the order of 10^6 variables
  - Converge within ~30 Newton iterations
    - Regardless of problem size!
    - Clear gains for large problems

114

6.1.1 Applications of NLP

- All previously examined formulations may result into NLP's to solve them
- Capabilities of NLP solvers nowadays can reach, with IPM, up to ~10^6 variables
- For convex problems there are reports of up to ~10^9 variables
- Not only can optimization be used off-line, but on-line optimization is possible – next application

- Model Predictive Control (MPC)
- Applications
  - Fast reliable online (real-time) optimizing controllers
  - Supply chain management problems
  - "Revenue management"
- Formulation
  - Output: control actions at each time instant
  - Input: current state of the system
  - Can handle disturbances (uncertainty component)
  - Objective: a mixture of control and economic criteria



- Starting at current time, produce control actions for the next *T steps:*

$$\min_{x(\tau),u(\tau)} \sum_{\tau=t}^{T-1} l(x(\tau),u(\tau)) \text{ convex (quadratic) objective}$$

subject to:

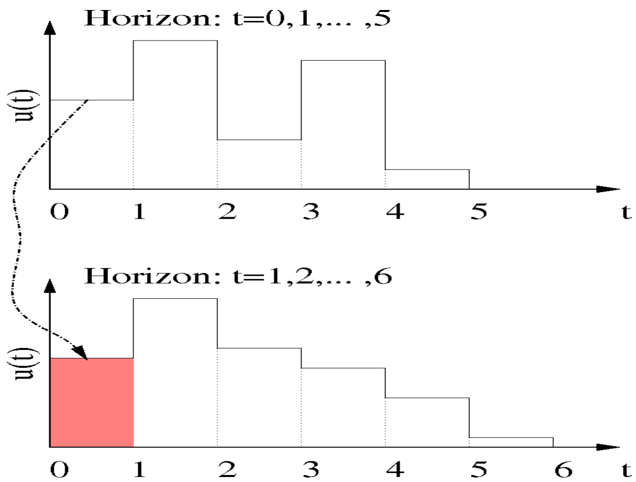
$$x(\tau+1) = A(\tau)x(\tau) + B(\tau)u(\tau) + \hat{d}(\tau)$$

$$x^L \le x(\tau) \le x^U$$

$$u^L \le u(\tau) \le u^U$$

$$\tau = t, t+1, t+2, \dots, t+T-1$$

$$x(t) = x_0$$



6.1.2 NLP people…

Lorenz T. Biegler

Bayer Professor of Chemical Engineering
Department of Chemical Engineering
Carnegie Mellon University

Specialization

Very large scale NLP, OCP, Parameter Estimation

Andreas Wächter

Research Staff Member (Nonlinear Optimization)

Thomas J. Watson Research Center, Yorktown Heights, NY USA

Specialization

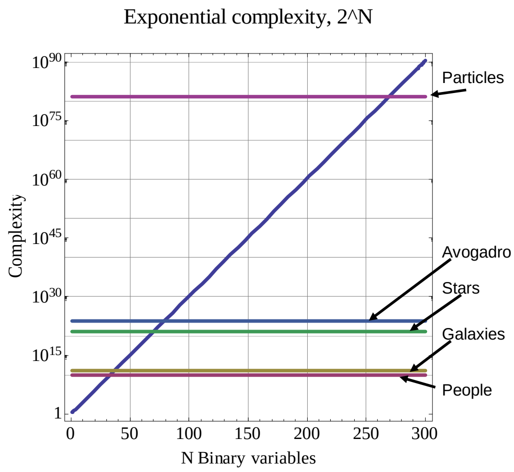
Very large scale NLP, developed IPOPT with L.T. Biegler

Mixed-Integer Nonlinear Programming (MINLP) solvers

6.2 Integer Programming (IP)

- Integer variables arise in many very important areas of industrial interest
  - Either for counting whole numbers
  - Or to capture embedded logic in mathematical process models
- Addressed early on, as soon as LP solvers matured
  - Mixed-Integer Linear Programming (MILP) models and solvers

- Will focus here on binary variables, {0,1}
  - General integer variables handled similarly
- Problem complexity → $O(2^N)$
  - Exponential explosion, combinatorial problems
  - Explicit enumeration only possible for tiny problems
- Solved via Branch and Bound (B&B)
  - *Implicit enumeration* method



- Why rounding up relaxed LP does not work; consider simple MIP problem

$$\min_x \ z = -(x + 5y)$$

$$subject\ to:$$

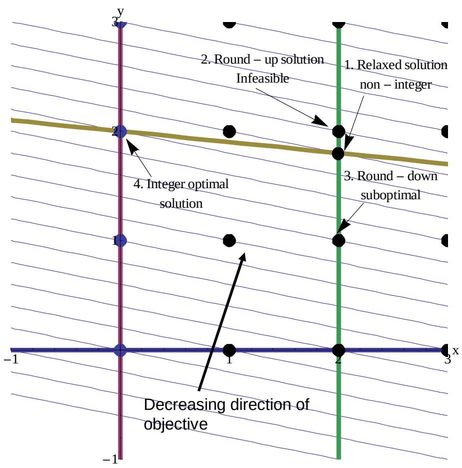
$$x + 10y \le 20$$

$$x \le 2$$

$$x, y \ge 0$$

$$x, y \in Integer$$

Adopted from:
Practical Optimization: a Gentle Introduction
John W. Chinneck, 2010
http://www.sce.carleton.ca/faculty/chinneck/po.html



For what will follow in this section, keep in mind 2 things:

1. Adding a constraint to an optimization problem
   – Will either do nothing to the solution (a loose constraint),
   – Or, will actively constrain the problem so that new optimum is worse than previous one
2. Fixing a variable to a given value
   – Will either do nothing (if at optimum…)
   – Or, will result in a worse optimum (fixing a value is like adding an equality constraint)

127

- IP example

$$\min_{x} \; z = -\left(8y_1 + 11y_2 + 6y_3 + 4y_4\right)$$

$$\textit{subject to}:$$

$$5y_1 + 7y_2 + 4y_3 + 3y_4 \le 14$$

$$y \in \{0.1\}^4$$

Adopted from:
Michael Trick's Operations Research Page, Associate Dean, Research and Professor, Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA USA 15213, http://mat.gsia.cmu.edu/orclass/integer/node13.html

128

- We begin by solving the *relaxed* LP

$$\min_{x} \; z = -\left(8y_1 + 11y_2 + 6y_3 + 4y_4\right)$$

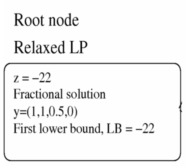
$$\textit{subject to}:$$

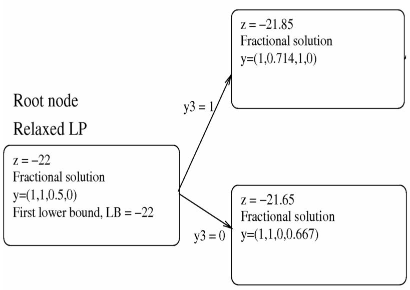
$$5y_1 + 7y_2 + 4y_3 + 3y_4 \le 14$$

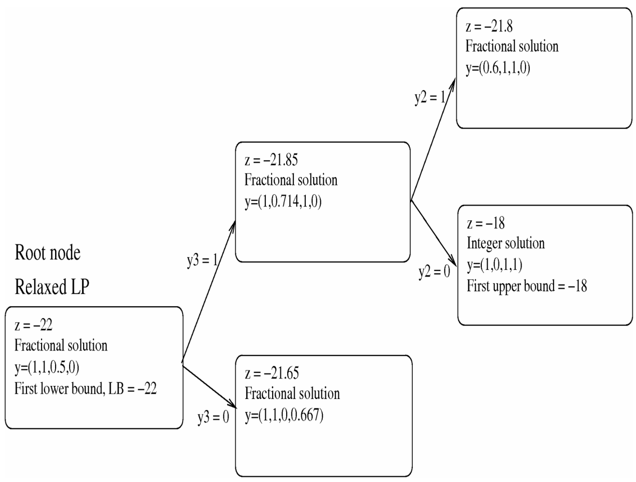
$$0 \le y_i \le 1; \quad i = 1,2,3,4$$

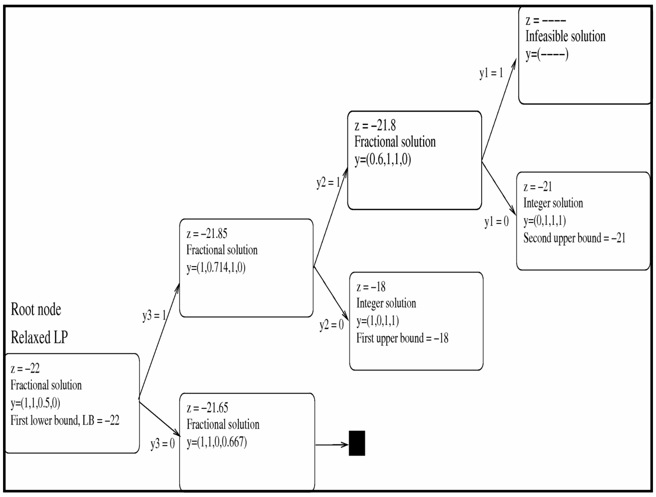
- Integrality constraints have been replaced by continuous bounds in the range 0-1

129









- Branch and bound
  1. Can prove global optimality of a solution
  2. Can be terminated in fewer iterations than complete proof of global optimality to save CPU cost
     - Tolerance on difference of bounds provided
  3. Provides rigorous bounds for the solution
  4. Usually terminates before it becomes exhaustive
     - Good formulations and constraint additions (cuts) help in this
  5. Can be used on MINLP as well as MILP
     - Usually direct B&B on NLP is not done (too expensive)
     - Special approaches exist using iterations of MILP and NLP subproblems
     - Cannot guarantee solution of nonconvex MINLP's and may fail often

134

6.2.1 Applications of IP

- Major applications of IP can be classified in 3 main areas

1. Process Synthesis
2. Scheduling
3. Transportation problems

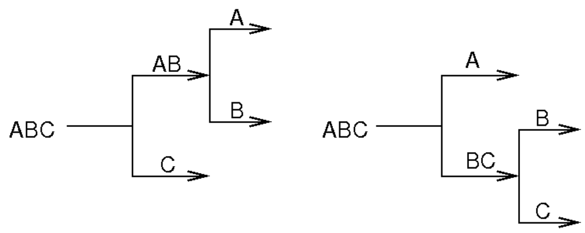
135

6.2.1.1 Process Synthesis

- Key idea
  – Include a sufficient number of alternative
    - Processes
    - Units
    - Interconnections
  – Have binary and continuous variables decide
    - Which is in and which is out
    - Connectivity
    - Operating conditions
  – The overall model is called a *superstructure*

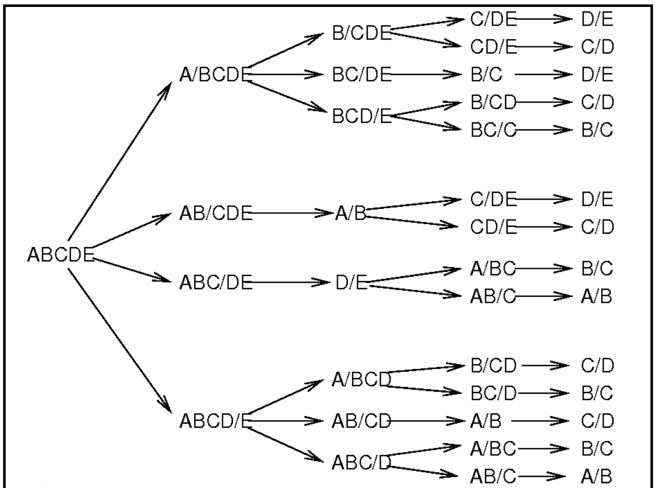
136

- Example: synthesis of multicomponent distillation sequences



Two possible separation sequences for a 3−Component mixture.

137



- Combinatorial complexity of the synthesis task

# Components	# Separations	# Sequences
N	$\frac{(N-1)N(N+1)}{6}$	$\frac{2((N-1))!}{N!(N-1)!}$
2	1	1
3	4	2
4	10	5
5	20	14
10	165	4862
20	1,330	1.77E+09
30	4,495	1.00E+15

- Example: Heat Integration

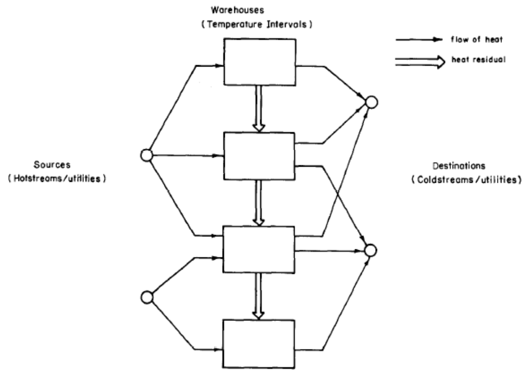


Fig. 2. Analogy of heat recovery network with transshipment model.

- Image taken from

  Reference:

  Papoulias, S.A., Grossmann, I.E., "A Structural Optimization Approach in Process Synthesis-II Heat Recovery Systems", Computers and Chemical Engineering, 7(6), 707-721, (1983).

- Formulated as a transportation (transshipment) problem
  - Leads to LP/MILP formulations
  - Completely equivalent to Pinch Analysis
  - More flexible (forbidden matches)

- HEN network design:
  - Formulated as a MINLP problem
    - All possible stream splits
    - All possible stream mixes
    - All possible bypasses
    - Nonconvex problem in general

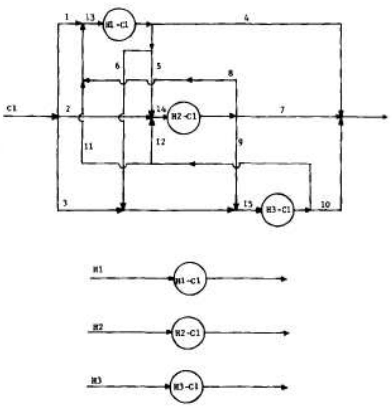


Figure 2. Stream superstructures of *C1, H1, H2, H3.*

- Image taken from

  Reference:

  Floudas C.A., A.R. Ciric and I.E. Grossmann, "Automatic Synthesis of Optimum Heat Exchanger Network Configurations", AIChE Journal, 32, 276 (1986).

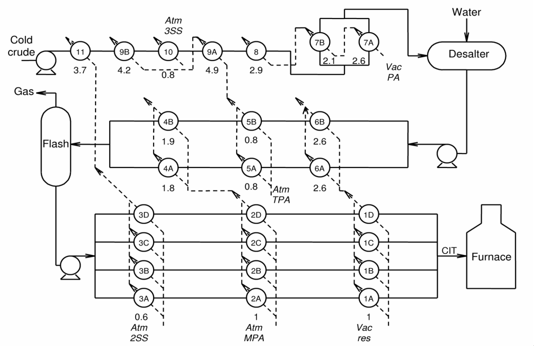
6.2.1.2 Scheduling

- Key ideas
  - Multiperiod operations
  - Binary variables (on/off) with indexing
    - {time, unit, process}, *etc.*
  - Continuous or integer variables for quantities
    - Flows
    - Inventories

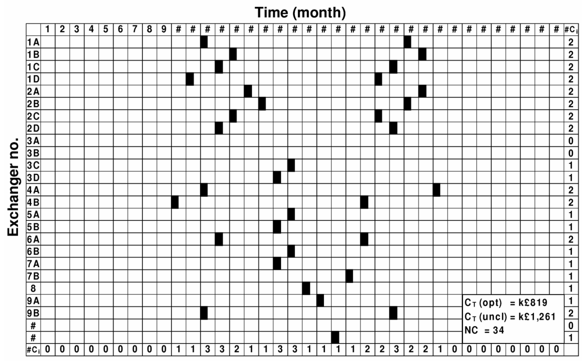
- Key areas

1. Batch scheduling, production planning
2. Supply-chain management
3. Scheduling of maintenance operations

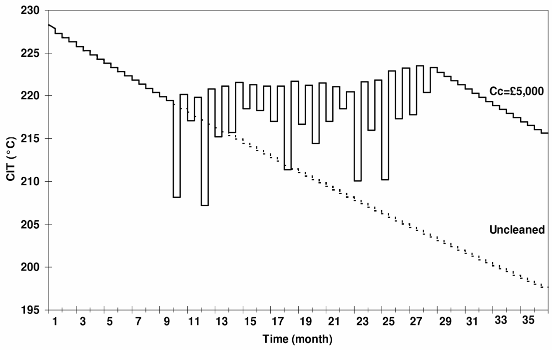
- Example: Scheduling of cleaning actions in HEN's subject to fouling



- Cleaning actions



- Crude inlet temperature



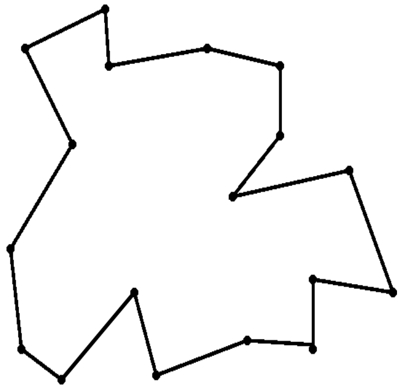
- *Reference:*
  Smaili, F., Vassiliadis, V. S., and D. I., Wilson, "Long-Term Scheduling of Cleaning of Heat Exchanger Networks: Comparison of MINLP/Outer Approximation based Solutions with a Backtracking Threshold Accepting Algorithm", Chem. Eng. Res. Des., Trans. IChemE, 80 (A6), 561–578, September (2002).

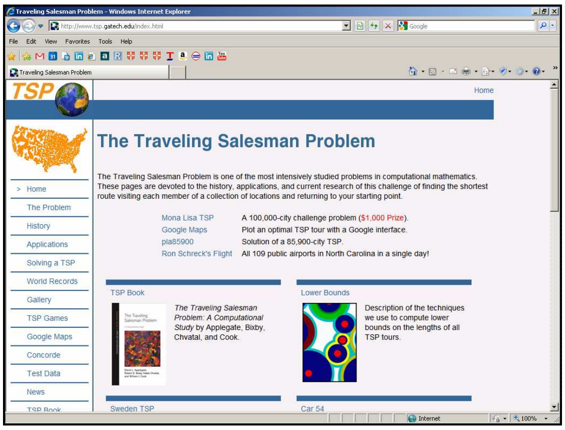
6.2.1.3 Transportation problems

- Key ideas
  - Belong to the area of network problems
  - Binary variables (on/off) with indexing
    - Routing logic: {start, destination}
  - Continuous or integer variables for quantities
  - Can be multiperiod operations
  - Can be multivehicle problems
- Key areas
  - Travelling Salesman Problem (TSP)
  - Vehicle Routing Problem (VRP)

- Example: TSP

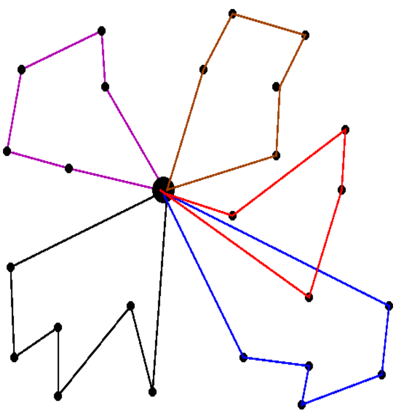
1. List of cities
2. Start at some city
3. Visit all cities once
4. Return to starting city
   - Closed circuit tour



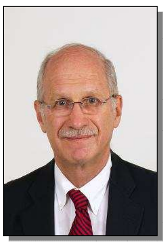


- Example: VRP

1. List of delivery locations
2. Allocate tour to each vehicle
3. Capacity constraints for each vehicle
4. Time windows of delivery
5. Start at depot
6. Return to depot



6.2.2 IP people…



Ignacio E. Grossmann

NAE member

Rudolph R. and Florence Dean University Professor

Department of Chemical Engineering

Carnegie Mellon University

Specialization

Large scale MILP and MINLP, formulation and solution methods

6.3 Deterministic Global Optimization

- We will focus on NLP problems
- Key ideas:
  - Intervals (bounds) for the values of all variables
  - Construction of convex underestimators of nonlinear (nonconvex) functions
  - Estimation of lower bounds of the NLP
  - Construction of a B&B tree based on upper and lower bounds of the NLP
  - Fathoming of nodes with LB > UB
  - Bisection of each variable interval at a time

157

- Original NLP

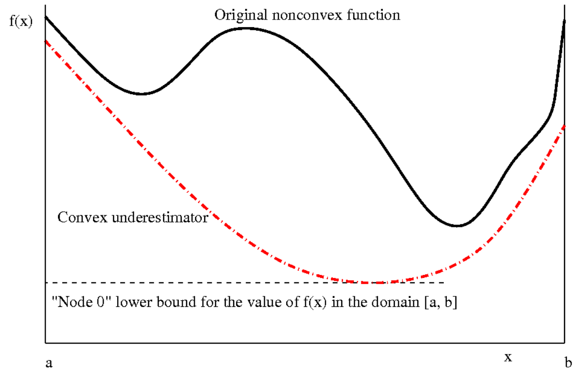
$$\min_x f(x)$$
$$subject\ to:$$
$$h(x)=0$$
$$g(x)\le 0$$
$$x^L \le x \le x^U$$

- Relaxation

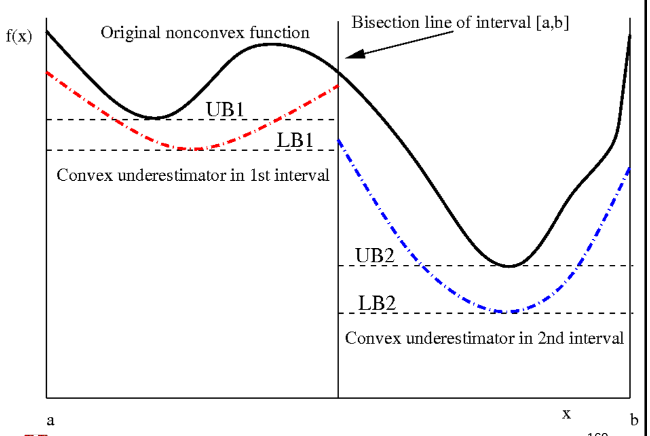
$$\min_x \underline{f}(x)$$
$$subject\ to:$$
$$\underline{h}(x)\le 0,\ \overline{h}(x)\ge 0$$
$$\underline{g}(x)\le 0$$
$$x^L \le x \le x^U$$

158

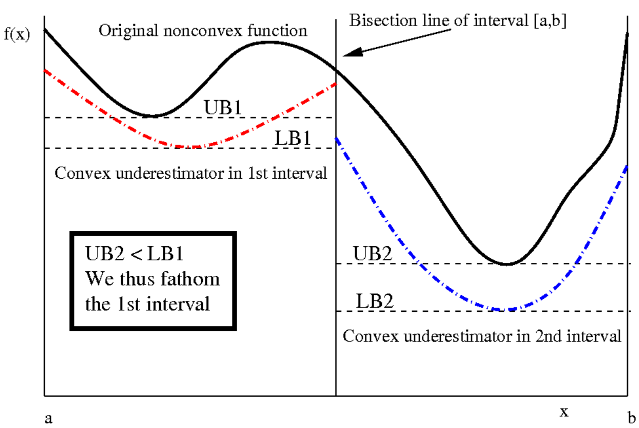
- Graphical example



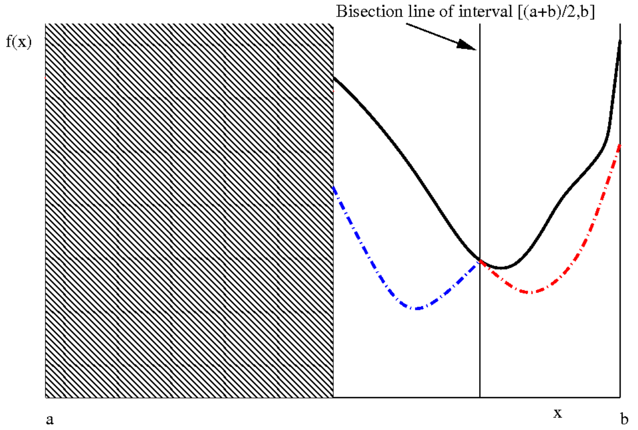
159



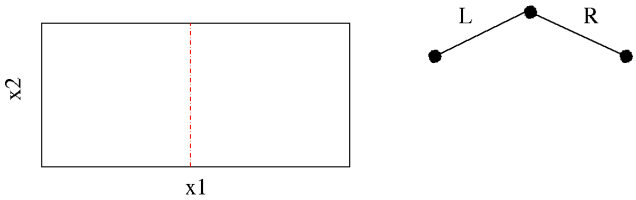
160

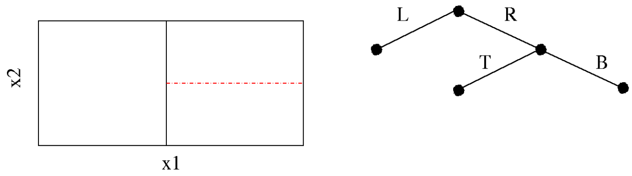


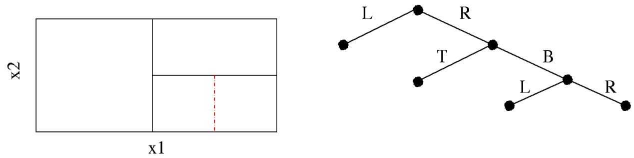
161

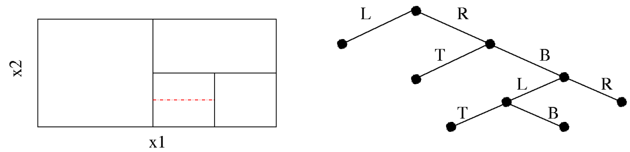


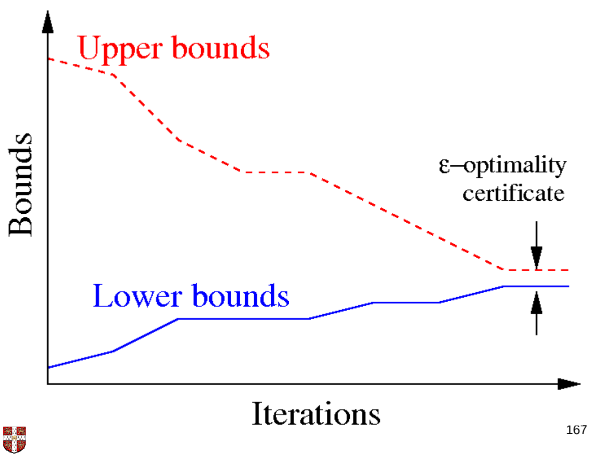
162





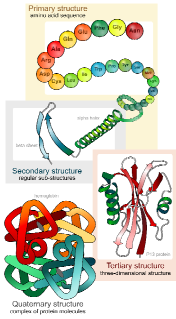




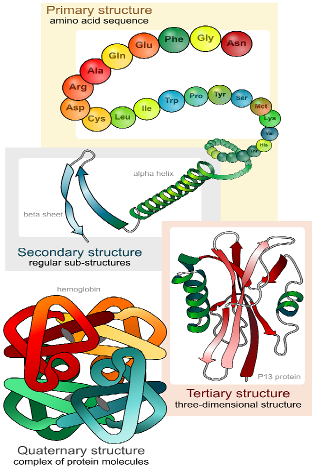


6.3.1 Protein Folding

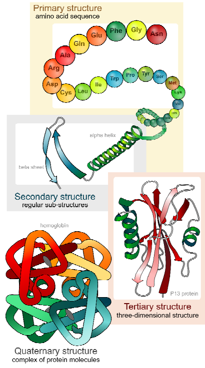
- Prediction of 3D structure of proteins
- The "holy grail" in global optimization
- Proteins are linear polymers of amino acids
- 3D structure determines function



- Structure determined by minimization of potential energy
  1. bending energy
  2. bond stretching energy
  3. bond torsion energy
  4. electrostatic energies on amino acids
- Exponential number of local minima
  – Number of amino acids



- Only a small proteins can be solved to guaranteed global optimality
- Capabilities of deterministic global solvers
  – Depend on problem size
  – Particularly on number of nonconvex terms involved
- A mixture of approaches is thus used this highly nonconvex problem
- Great international research interest in deterministic global optimization
  – *Ab initio* structure prediction
  – *De novo* design of proteins



6.3.2 Deterministic Global Optimization people…

Christodoulos A. Floudas

NAE member
Stephen C. Macaleer '63 Professor in Engineering and Applied Science
Professor of Chemical and Biological Engineering
Department of Chemical and Biological Engineering at Princeton University

Specialization

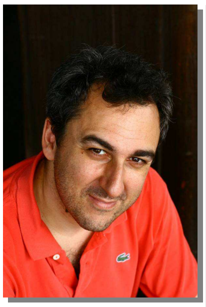
Nonconvex NLP and MINLP, formulation and solution methods
Protein folding

Nikolaos V. Sahinidis

John E. Swearingen Professor of Chemical Engineering
Department of Chemical Engineering
Carnegie Mellon University

Specialization

Nonconvex NLP and MINLP, formulation and solution methods



Leo Liberti

Professeur Chargé de Cours
LIX, Ecole Polytechnique, Palaiseau, France

Specialization

Nonconvex optimization, combinatorial optimization

Sven Leyffer

Mathematics and Computer Science Division at Argonne National Laboratory

Specialization

Theory and applications of NLP, MINLP and global optimization

"God does not care about our mathematical difficulties. He integrates empirically."

Albert Einstein

7. Derivative-free optimization

"I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right."

Albert Einstein

- Other names for these methods
  - Pattern Search (PS) methods
  - Direct Search methods
  - Derivative-free methods

- Very old methods (1950's)
  - Depend only on function evaluations
  - Do not need gradients
  - Can deal with nonconvex and discontinuous functions
  - Generally very robust, *i.e.* don't crash!
  - Need many function evaluations
  - Solve:

$$\min_x f(x)$$
$$subject\ to:$$
$$x^L \leq x \leq x^U$$

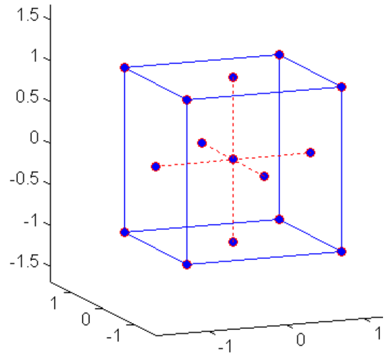
- Resurgence in recent years
  - Large scale application example: fitting (training) of neural networks (NN)
- Examples
  - Nelder Mead method (amoeba)
    - Storage $O(N^2)$
    - 1 function evaluation per iteration
    - N+1 function evaluations to start
  - Cyclic Coordinate Search (CCD)
    - Classic method
    - Storage $O(N)$
    - 2N function evaluation per iteration
    - $2N$+1 function evaluations to start

7.1 The CCD method

- Idea
  - Search cyclically each coordinate in an up and down step
  - When no better point found reduce stepsize
- Properties
  - The simplest direct search method
  - Many function evaluations in its classical incarnation
  - Severely scale-dependent,
    - Worse convergence than Steepest Descent

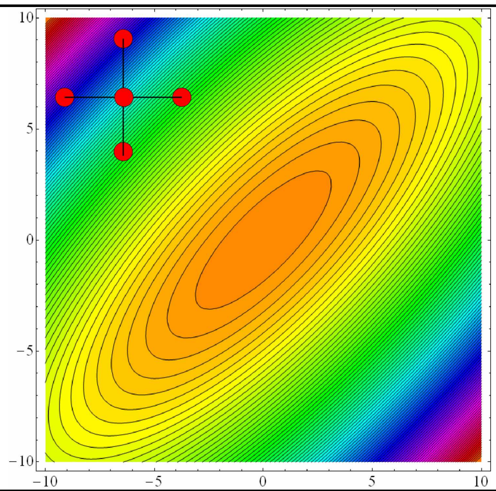
181

- Identical to the fractional factorial experiment design method

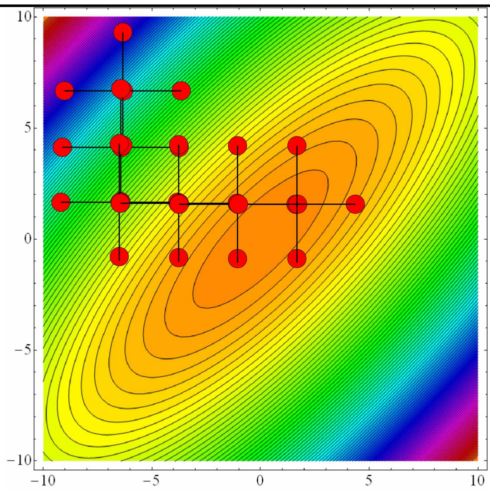


182

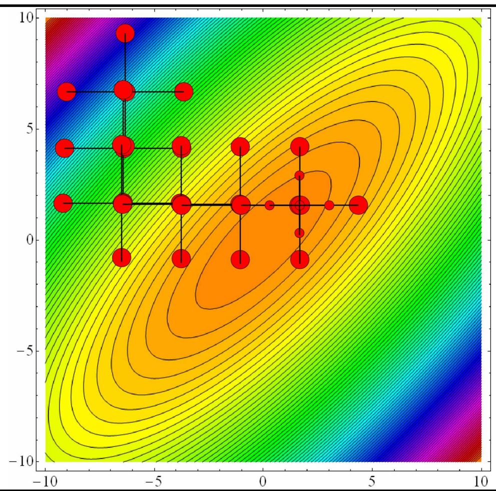
start



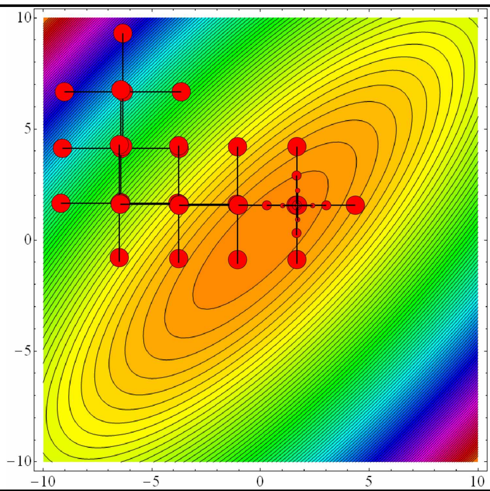
183



184

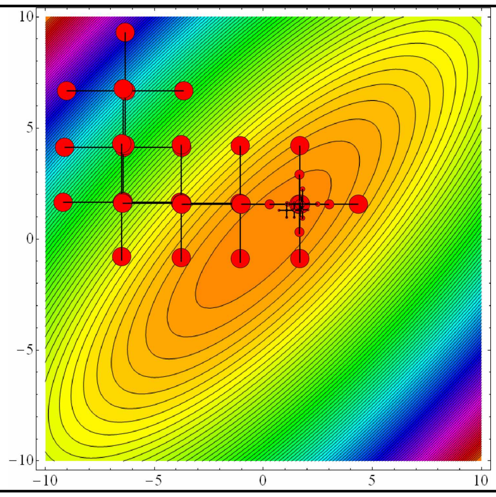


185



186

end



187

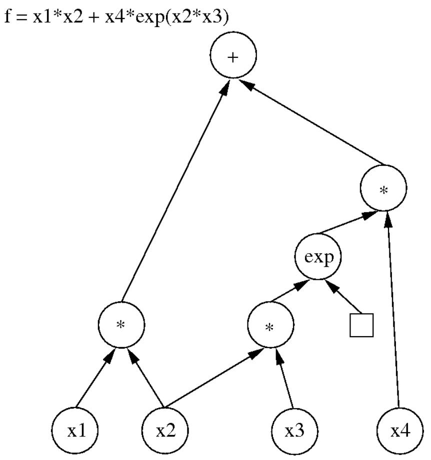
7.1.1 Speeding up the CCD

- First we need to see how functions are represented in computer code (*factorable functions)*
- Arithmetic evaluation trees
  - Directed Acyclic Graph (DAG)
- Used in Automatic Differentiation, where

$$O(\nabla_x f(x)) \leq 4 \cdot O(f(x))$$

for any number of variables

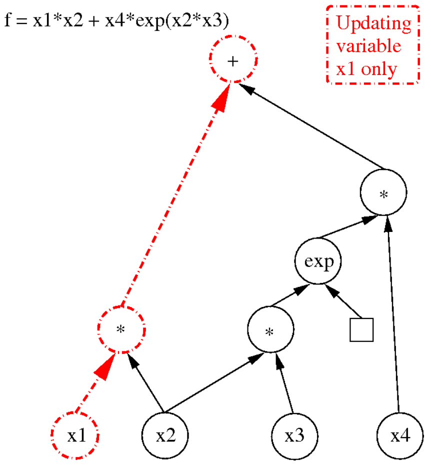
188



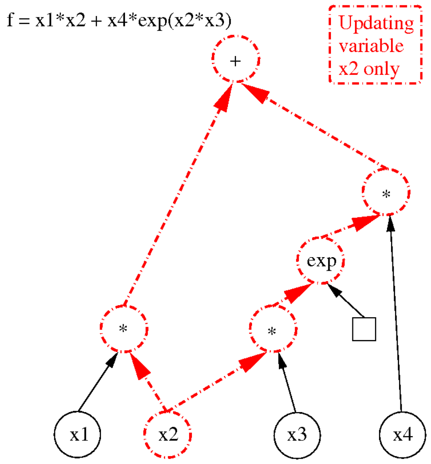
189

- Idea:
  - Since the CCD performs a variable-at-a-time perturbations around the base point
  - Store intermediate evaluations in the tree during every evaluation
  - Update only branches that change subject to a variable change

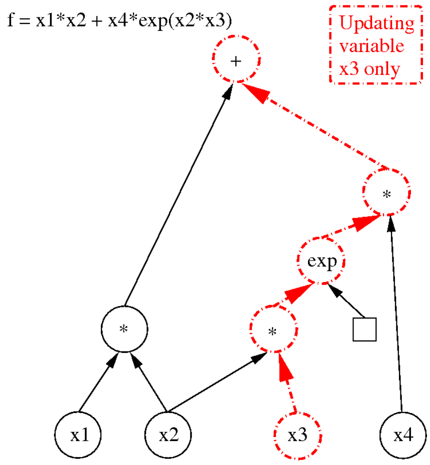
190



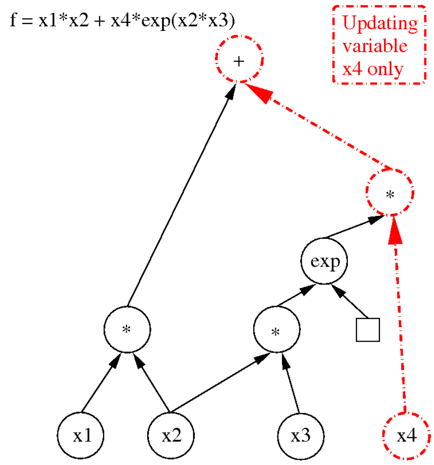
191



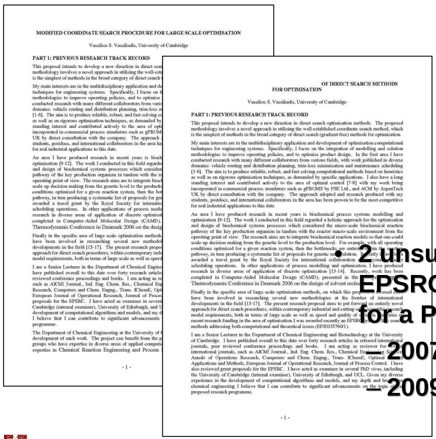
192



193

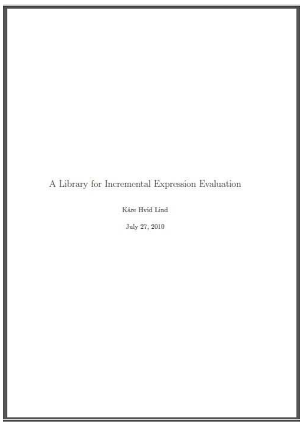


194



- 2 unsuccessful EPSRC proposals for a Ph.D. project
  - 2007
  - 2009

195

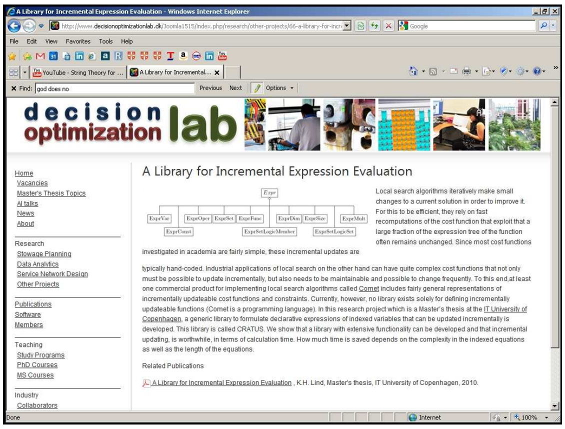

A Library for Incremental Expression Evaluation

Kåre Hvid Lind

July 27, 2010

- 2010 Copenhagen (IT University) MSc Thesis
  - Coding of function DAG speedup ideas
  - Fortunately no work on the other topics identified in our earlier unsuccessful proposals
- Commercial software application made available by them though

196



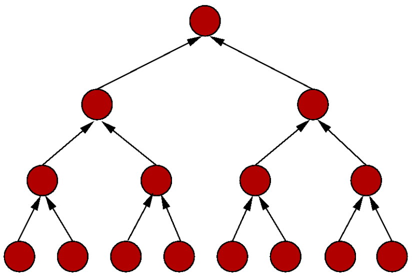
197

- Theoretical indicators of function evaluation speedups
  - For a full cycle of the classic CCD method
  - Operation counts and speedup ratios for
    1. Balanced binary tree function types
    2. Sum of bilinear terms

198

BALANCED BINARY TREE

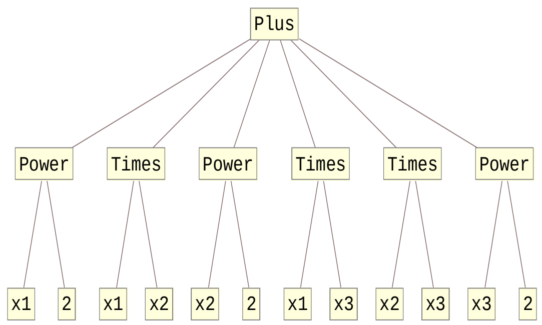
Total Nodes = 2^(NL+1) – 1;   Inner Nodes = 2^NL–1

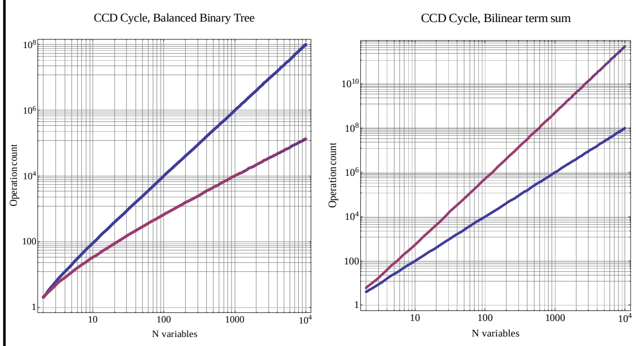


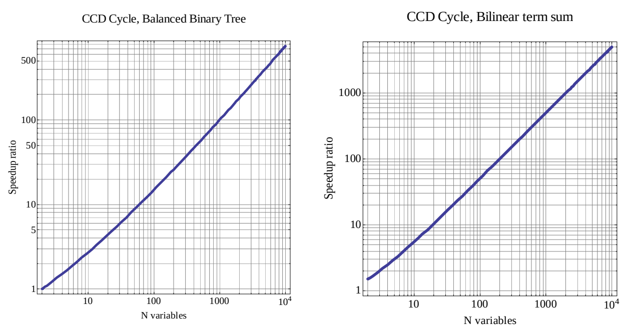
Level	Nodes	Total
0	1	1
1	2	3
2	4	7
3	8	15

- Sum of pure bilinear terms

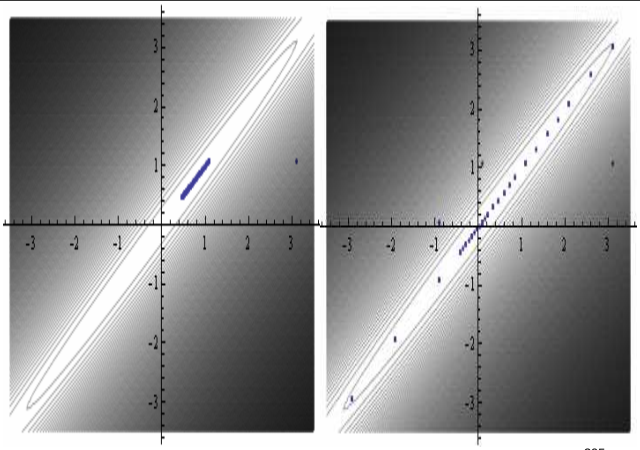
$$f(x) = \sum_{i=1}^{N} \sum_{j=i}^{N} x_i x_j$$







- Further work to be done:
  - Does not matter even if we make CCD 1000's of times faster
    1. Must ensure we can use it for high dimensionality
    2. Explore inclusion of constraints in a useful way
    3. Above all, deal with the severe scaling limitations
       - Next and final slide



205

Albert Einstein on research…

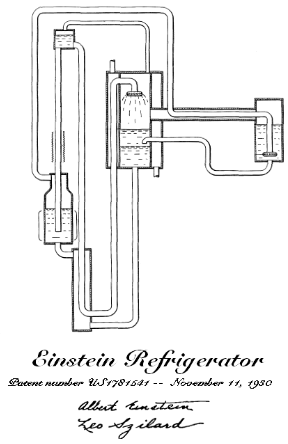
- To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science.
- If we knew what it was we were doing, it would not be called research, would it?
- Science is a wonderful thing if one does not have to earn one's living at it.
- If the facts don't fit the theory, change the facts.

206

Mission of research

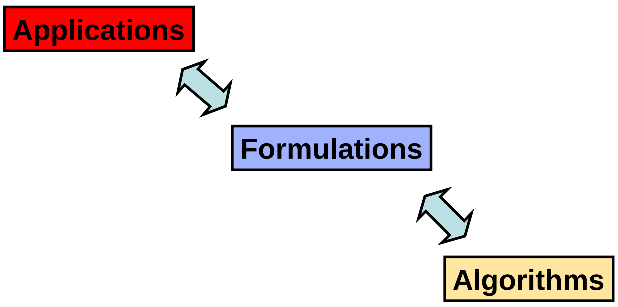
- Create the climate for new technologies and theories to emerge, advancement of society
- Train new researchers for the future, on a new topic, *i.e.* PhD students
- Train more experienced researchers in an advanced topic, a type of finishing school, *i.e.* postdoctoral researchers

207

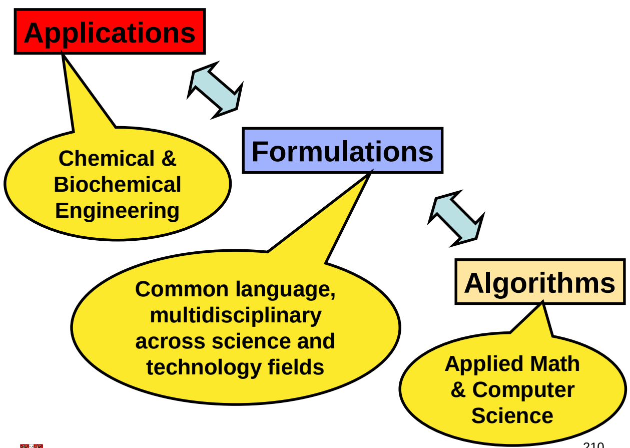


208

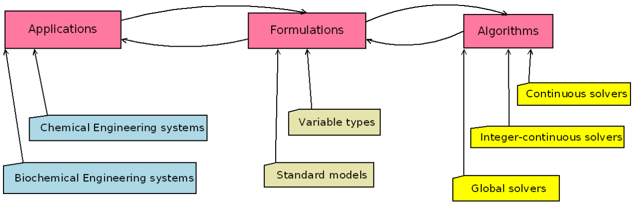
8. Conclusions



209



210



211

The optimization alphabet soup

MP IP MIP

LP NLP CP

MILP MINLP

CVP DAG MPC

OCP DYNOPT MOO BLP SP

SIMPLEX IPM CCD PS

212

Stephen P. Boyd

Stanford University

Samsung Professor in the School of Engineering

Professor, Information Systems Laboratory, Department of Electrical Engineering,

Professor (by courtesy), Department of Management Science and Engineering

Institute for Computational and Mathematical Engineering

Specialization

•Convex Optimization

213

- Convex optimization is the basis of all theory and methods
- The following is an outstanding book:

  Convex Optimization
  Stephen Boyd and Lieven Vandenberghe
  Cambridge University Press, 2004

214

"The important thing is not to stop questioning."
Albert Einstein

Friday 6 May 2011 / Part 1/2 / 3.30pm / Lecture Theatre 1
Friday 13 May 2011 / Part 2/2 / 3.30pm / Lecture Theatre 1

Optimisation: Formulations, Algorithms and Applications (…but no algebraic spaghetti)

Dr. Vassilis S. Vassiliadis

Department of Chemical Engineering and Biotechnology
University of Cambridge
Pembroke Street
Cambridge CB2 3RA

Abstract

Optimization is, simply put, the science of finding the best solution amongst many feasible alternatives for general decision making problems. Every engineer and scientist will most certainly have encountered optimization in some form or another: from parameter estimation and model fitting, to experiment design, and to more advanced uses, such as optimising processes and plant flow sheets, and more.

A brief search through the Web will verify that there is an enormous volume of publications and books on the subject, regarding both applications and theoretical developments. There is no doubt that optimization theory can be very difficult to grasp, if looked at the level a mathematician would use to develop a mathematical proof.

However, this is not the intent of this presentation. The aim is to present optimization as an indispensable tool in modern engineering science. The intended audience is anyone interested to learn about optimization: where it can be applied in our discipline, how to formulate appropriate models, and where the state-of-the-art has reached with modern solver codes.

The level is such that the presentation will be accessible to undergraduate students at any year of the Tripos, whilst presenting the topic in a way that is useful to researchers as well. There will be no complex mathematics, but some equations will be used: basic algebra, basic calculus and a lot of common sense! Most of the ideas presented will be highlighted by applications in Chemical Engineering.

Vassilis S. Vassiliadis

Dr. Vassiliadis' research interests lie in the development and application of optimization and simulation algorithms in engineering and scientific domains. His research field is Process Systems Engineering, a sub-discipline within Chemical Engineering.

He obtained his Diploma in Chemical Engineering (M.Eng.) in the School of Chemical Engineering at the National Technical University of Athens in 1989, having graduated with distinction and top of his class. He then studied for his Ph.D. in Process Systems Engineering, in the Department of Chemical Engineering and Chemical Technology at Imperial College, London, from where he graduated in 1993. He then spent a year working as a postdoc in Princeton University.

He joined the Department of Chemical Engineering at Cambridge as an Assistant Lecturer in 1995 and is now a Senior Lecturer. He has acted as a consultant to AspenTech LTD for the development of an optimal control solver code, and his Ph.D. code for optimal control formed a prototype solver for gPROMS, the dynamic simulator by PSE LTD.