Constrained optimization methods in the design problems of thermal-hydraulic systems

Maciej Kulig

Overview of the presentation

- Introduction
- Formulation of the problem
  - Specific features of engineering design problems
  - Example of an engineering design problem
  - Practical applications – difficulties and challenges
- Numerical methods of optimization
  - Overview of the selected algorithms
  - IEA work on the implementation of suitable tools
  - Results of numerical experiments
- Achievements in the field of optimization methods
  - Brief overview of NLP methods currently available
  - Future prospects in the design optimization

Introduction

- Modelling and **solving design problems** using complex economic/technical criteria **needed in many fields**
- **Progress in solving complex optimization problems**
  - Growing capabilities/accessibility of digital technology
  - Achievements in mathematics & numerical methods
- **Numerical difficulties:**
  - **Complexity** of the mathematical model describing a system being optimized
  - **Calculation** of the model **laborious** & time consuming
  - **Truncation / round-off problems** make the optimization process more difficult

Institute of Nuclear Research/Institute of Atomic Energy, Świerk

- IBJ, Department of Reactor Engineering, E-IX, 1962/69 – studies related to the planning of nuclear power programme, including optimization methods
- Numerical methods for optimization – (IAEA fellowship, University of Birmingham, 1969/70)
- IEA, Design Department, 1977/81 – analyses of thermal-hydraulic systems (practical applications: reactor for district heating, modernization of research reactor EWA)
- Cooperation with other organizations/projects (Warsaw University of Technology, CHEMADEX)

Part 1

Formulation of the problem

Specific features of thermal hydraulic problems

Optimization model

- **Objective function** – combination of indicators of technical and/or economic nature that describe the goal
- **State equations** – representing general laws of physics (mass & energy balance, conservation of momentum, heat transfer conditions,…)
- **Constraints** – Technical requiremnts (technological, structural, thermal, etc.) or physical (related to the physical meaning of the variable, limitations of the model, etc.)

Constructing the model

- **Balance nodes** - distinct parts of the system (elements, devices or junctions of the system) encompassed within the balance barrier
- **Inter-node connections** – used for modelling energy and/or mass transfer through the balance barrier
- **State variables** – characterize inter-node connections (thermodynamics parameters)
- **Node characteristics** – design parameters of subsystems/devices

enco

Mathematical formulation of the problem

Minimize a scalar function $F(\underline{x})$,

$$F(\underline{x}^*) = \min_{\underline{x} \in G} F(\underline{x})$$

$$G = \{ \underline{x} \in R^n : \underline{\varphi}(\underline{x}) \leq 0,\ \underline{\psi}(\underline{x}) = 0 \}$$

Where:

$\underline{\varphi}(\underline{x}) = \varphi_i \qquad i = 1,2,\ldots,m_1$

$\underline{\psi}(\underline{x}) = \psi_i \qquad i = 1,2,\ldots,m_2$

$F,\ \varphi_i,\ \psi_i : \qquad \mathbf{R}^n \rightarrow \mathbf{R}$

Comment: For the design optimization problems $m_2 < n$;
$n - m_2$ – the degree of freedom

Specific features of thermal hydraulic problems

Difficulties

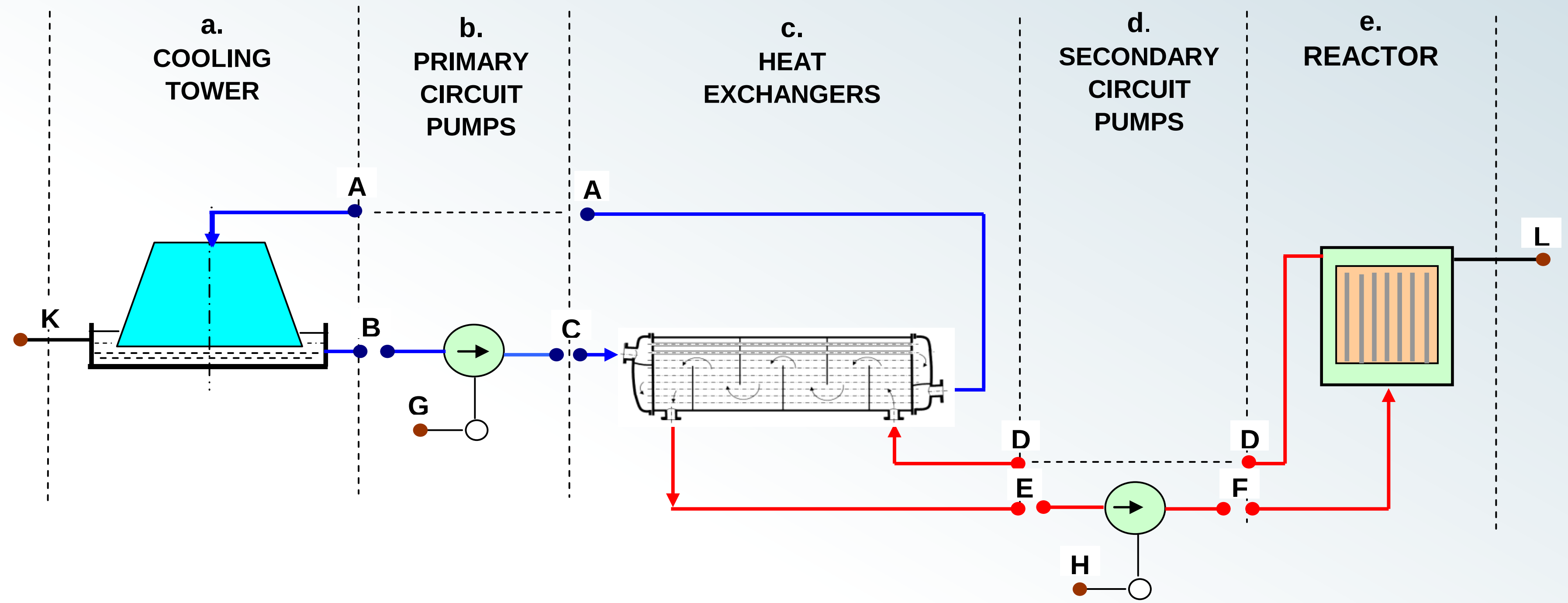
- **Complex & nonlinear relationships** for the state equations, objective function F and constraints φ, ψ –
  - **Large number of nodes** (i.e. # of equations & state variables),
  - **State equations**, **objective** F, **constraints** φ, ψ (majority) **nonlinear**
- **Significant execution time** needed for calculating F, φ, ψ
- Desirable **regularity conditions** of the problem (e.g. the convexity of functions) impossibile to ascertain
- **Analytical representation of the gradients** for the functions F, φ, ψ **practically imposibile**; calculating the gradients requires approximation by finite differences

Specific features of thermal hydraulic problems

Features that can help making the problem easier to solve

- **Structure of the model** – inter-node connections limited (Jacobian of the state equations has a strip or triangle structure)
- **Part of the state equations** linear with regard to certain variables (then, the equality constraints can be used to eliminate variables)
- **Proper formulation of the problem** helps to reduce the effort for calculating the model F, φ, ψ (therefore, also the effort of finding the solution)
  - **Possibility to eliminate significant number of equality** constraints (state equations) reducing the number of decision variables (therefore, also the effort of finding the solution)
  - **Posibility to achieve a block structure of the algorithm** for calculating F, φ, ψ (reduces the effort for numerical approximation of the gradients)

Example design problem
(modernisation of the research reactor „EWA”)



Reactor cooling system - definition of balance nodes;
A, B, …, F - nodes connections; G, H, K, L - outside connection (environment)

Example design problem (research reactor „EWA”)

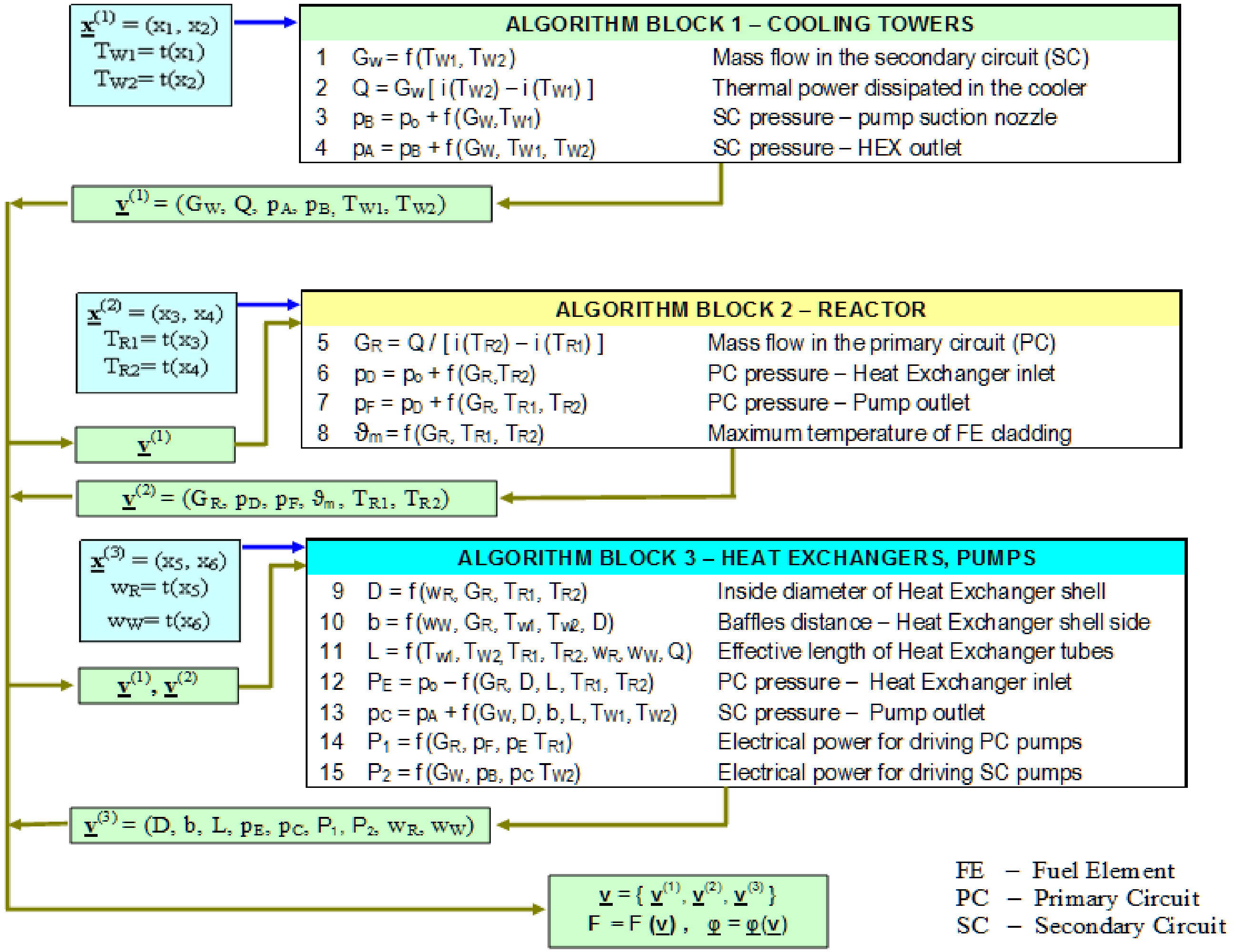
Definition of the state equations for the problem

NODE	Number and type of the equation		Variable number & name / Form of the relationship	Cooling Tower						Reactor						Heat Exchangers							Pumps	
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
				T_{W1}	T_{W2}	G_W	Q	p_B	p_A	T_{R1}	T_{R2}	G_R	p_D	p_F	ϑ_m	w_R	w_W	D	b	L	p_C	p_E	P_1	P_2
COOLER	1	Heat transfer	$f(G_W, T_{W1}, T_{W2}) = 0$	x	x	x																		
	2	Heat balance	$Q - G_W[\, i(T_{W2}) - i(T_{W1})\,] = 0$	x	x	o	o																	
	3	Flow	$p_B - p_0 - f(G_W, T_{W1}) = 0$	x		x		o																
	4	Flow	$p_A - p_B - f(G_W, T_{W1}, T_{W2}) = 0$	x	x	x		o	o															
REACTOR	5	Heat balance	$Q - G_R[\, i(T_{R2}) - i(T_{R1})\,] = 0$				o			x	x	o												
	6	Flow	$p_D - p_0 - f(G_R, T_{R2}) = 0$								x	x	o											
	7	Flow	$p_F - p_D - f(G_R, T_{R1}, T_{R2}) = 0$							x	x	x	o	o										
	8	Heat transfer	$\vartheta_m - f(G_R, T_{R1}, T_{R2}) = 0$							x	x	x			o									
HEAT EXCHANGERS	9	Continuity	$w_R - f(G_R, D, T_{R1}, T_{R2}) = 0$							x	x	x				o		o						
	10	Continuity	$w_W - f(G_R, D, b, T_{W1}, T_{W2}) = 0$	x	x	x											o	o	o					
	11	Heat transfer	$Q - f(T_{W1}, T_{W2}, T_{R1}, T_{R2}, w_R, w_W, L) = 0$	x	x		o			x	x					x	x			o				
	12	Flow	$p_D - p_E - f(G_R, D, L, T_{R1}, T_{R2}) = 0$							x	x	x	o					x	x	o		o		
	13	Flow	$p_C - p_A - f(G_W, D, L, b, T_{W1}, T_{W2}) = 0$	x	x	x												x	x	x	o			
PUMPS	14	Energy balance	$P_1 - f(G_R, p_F, p_E\ T_{R1}) = 0$							x		x		o								o	o	
	15	Energy balance	$P_2 - f(G_W, p_B, p_C\ T_{W2}) = 0$		x	x		o													o			o

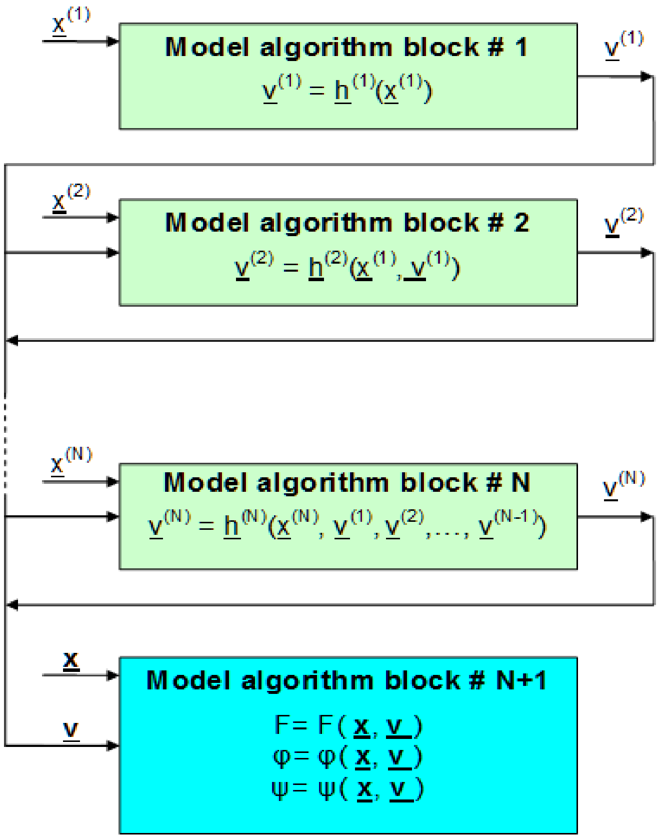
x NONLINEAR RELATIONSHIP   o LINEAR RELATIONSHIP

enco

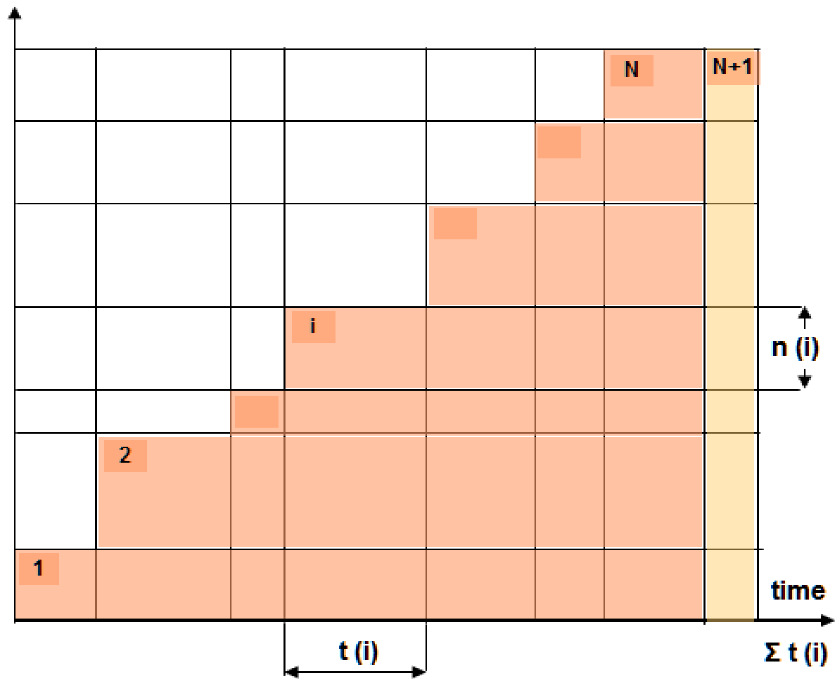
Research Reactor „EWA” - Algorithm for calculating the objective function & constraints – use of block structure of the state equations



Numerical approximation of gradients – benefit of block structure of the model algorithm



Numerical aproximation of gradient - computer time benefit when the model algorithm has a block structure



t (i) - Computer execution time for calculating block # "i"

n (i) - Number of variables associated with block # "i"

Comment: Benefits not limited to gradient aproximation; useful in calculating $F(\underline{x})$ in any new point $\underline{x} = \underline{x}_o + \underline{e}_i \Delta_x$

Part 2

Numerical methods of optimization

Numerical methods for constrained optimization (ConOpt)

- In general, optimization techniques can be divided into two classes – **direct** and **indirect**
- **Direct methods** start at an arbitrary set of values of variables and **proceed** step by step towards optimum, **by successive improvements**
- **Indirect methods** taking into accont the maximum knowlegde about the objective function and the domain of fisibility, usually in the form of analytic functions and inequalities, **replace the original problem** by others easier to solve.
- **Direct methods are easier to program** and usually **prefered** by practitioners

Numerical methods for constrained optimization (ConOpt)

- **Direct methods:**

$$\underline{\mathbf{x}}^{(0)}, \underline{\mathbf{x}}^{(1)}, \underline{\mathbf{x}}^{(2)}, \ldots, \underline{\mathbf{x}}^{(k)}, \underline{\mathbf{x}}^{(k+1)}, \ldots, \underline{\mathbf{x}}^{*}$$

Each iteration $\underline{\mathbf{x}}^{(k)} \rightarrow \underline{\mathbf{x}}^{(k+1)}$ includes two elements:

1) **Exploration of the model** (values of functions $\boldsymbol{F}$, $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$, their gradients, sometimes their second derivatives) usually in close neighborhood of the last point $\underline{\mathbf{x}}^{(k)}$
2) **Correction of decision variables** $\underline{\mathbf{x}}^{(k)}$ based on the results of exploration (conducted according to the pre-determined 'plan'/algorithm)

Initially the methods have been developed for unconstrained optimization (UncOpt), and then extended for the problems with constraints (ConOpt)

Numerical methods for constrained optimization (cont.)

Two groups can be distinguished in the direct methods:

- **Those which use the modification of the objective function**

  Modified function includes „penalty" terms added as a constraint is approached; typically the original problem (ConOpt) is converted into a sequence of unconstrained optimizations (in some methods only one)

- **Methods in which the direction of search is modified without altering the function**

  Usually they attempt to follow the active constraint or try to 'rebound' from them and so continue the search in the feasible region

Numerical methods for constrained optimization (cont.)

'Parameric' penalty function methods

$$\Phi(\underline{x}, \underline{r}) = F(\underline{x}) + P(c(\underline{\mathbf{x}}), \underline{\mathbf{r}})$$

P – Suitably defined skalar function

c(x) - Constraints (equality $\psi(\underline{x}) = 0$ and inequality $\varphi(\underline{x}) \leq 0$)

$\underline{\mathbf{r}}$ – Vector of parameters that regulate 'penalizing' effect of *P*

If the values of variable $\underline{\mathbf{x}}$ that minimize $\Phi(\underline{\mathbf{x}}, \underline{r})$ is $\xi(\underline{r})$ then $\xi(\underline{\mathbf{r}})$ is the solution to the constrained problem:

minimize F($\underline{\mathbf{x}}$) subject to the constraints $c_j(\underline{\mathbf{x}}) = c_j[\xi(\underline{r})]$

Then, to find the solution of the original constrained problem it is necessary to obtain values of the parameters $\underline{\mathbf{r}}$ such that

$$\Psi_j[\xi(\underline{r})] = 0, \qquad j = 1, 2, \ldots, m_1,$$
$$\varphi_j[\xi(\underline{r})] \leq 0, \qquad j = m_1+1, m_1+2, \ldots, m$$

Numerical methods for constrained optimization (cont.)

- **Penalty function of Powell** (1968) for the problems with equality constraints ($\psi(\underline{x}) = 0$)

$$\Phi(\underline{x},\underline{\sigma},\underline{\Theta}) = F(\underline{x}) + [\psi(\underline{x}) + \underline{\Theta}\,]^T\, \underline{S}\, [\psi(\underline{x}) + \underline{\Theta}\,]$$

where: $\underline{\boldsymbol{\sigma}} = (\sigma_1\ ,\ \sigma_2\ ,\ \ldots,\ \sigma_m\ )$; $\underline{\boldsymbol{\Theta}} = (\Theta_1,\ \Theta_2,\ldots,\ \Theta_m\ )$ - parameters

$\underline{S}$ – diagonal matrix m x m with elements $\sigma_i > 0$

- If $\xi^{(k)}$ is the solution to the unconstrained problem of the function $\Phi(x,\ \underline{\sigma}^{(k)},\ \underline{\Theta}^{(k)})$ and $\sigma_i^{(k)}$, i=1,2,..., m are sufficiently large then the formulas:

$$\underline{\boldsymbol{\sigma}}^{(k+1)} = \underline{\boldsymbol{\sigma}}^{(k)} \qquad \text{and} \qquad \underline{\boldsymbol{\Theta}}^{(k+1)} = \underline{\boldsymbol{\Theta}}^{(k)} + \underline{\psi}(\xi^{(k)})$$

ensure linear convergence at as fast a rate as is required.

- If the convergence is too slow then:

$$\underline{\boldsymbol{\sigma}}^{(k+1)} = \omega\, \underline{\boldsymbol{\sigma}}^{(k)} \quad \text{and} \quad \underline{\boldsymbol{\Theta}}^{(k+1)} = \underline{\boldsymbol{\Theta}}^{(k)}/\omega \qquad \text{where } \omega > 1$$

Numerical methods for constrained optimization (cont.)

- **Powell's algorithm** (1968) - can be extended for problems with inequality constraints:
  - **By the use of additional variables** (‘*slack variables*’) and converting the inequality constraints:

$$\varphi_j(x_1,\dots,x_n) \leq 0 , \quad j=1,\dots, m_1,$$

into equality constraints $\psi(x) = \varphi_j(x_1,\dots,x_n) + x_{n+j}^2 = 0$;

  - **By the use of 'penalty' function** (Michalski, Szymanowski, 1970)

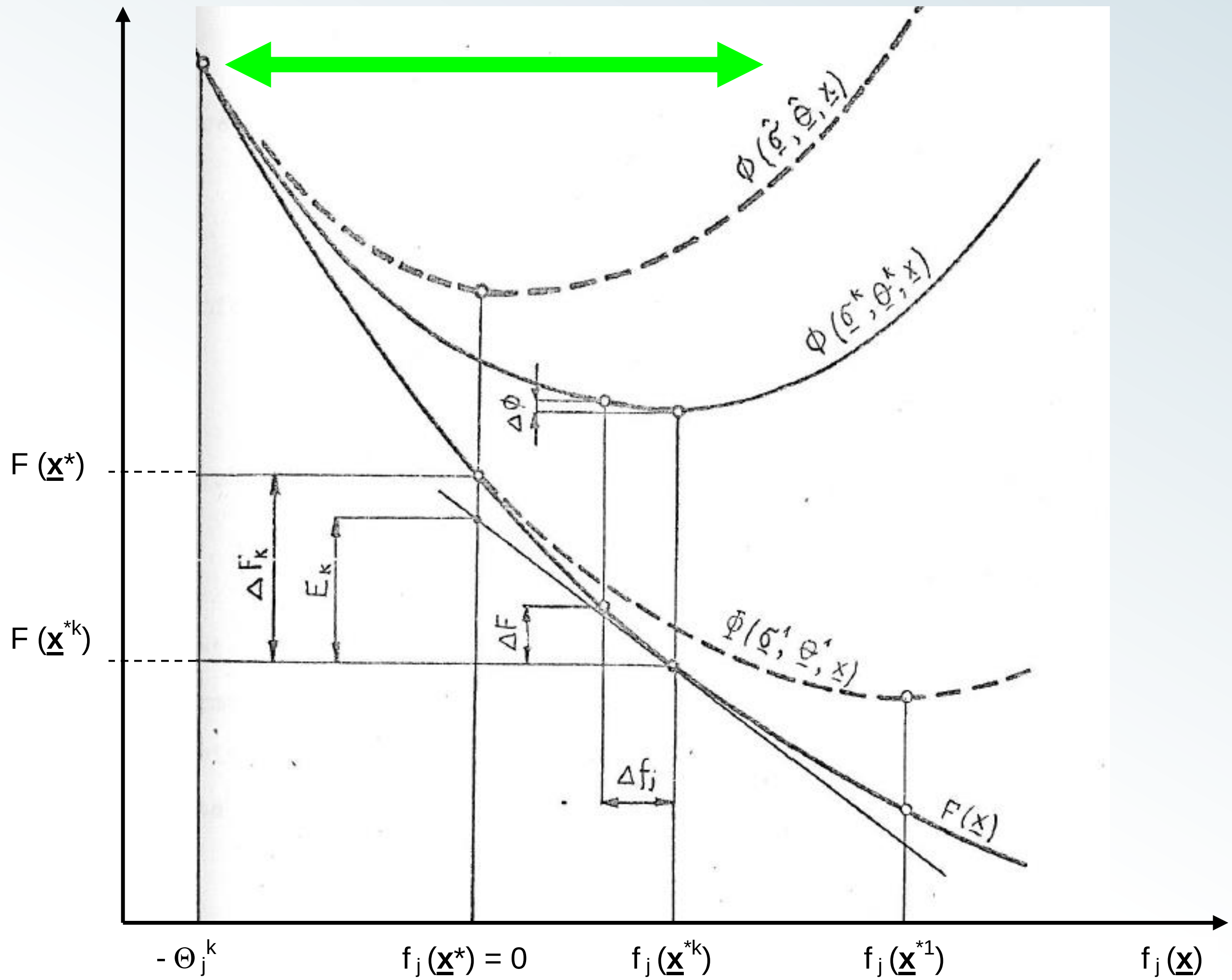
$$\Phi(\underline{x},\underline{\sigma},\underline{\Theta}) = F(\underline{x}) + \sum_{j=1}^{m} \sigma_j \, [\varphi_j(\underline{x}) + \Theta_j] \; \max\,[0, \varphi_j(\underline{x}) + \Theta_j]$$

where the formula: $\underline{\boldsymbol{\Theta}}^{(k+1)} = \underline{\boldsymbol{\Theta}}^{(k)} + \boldsymbol{\psi}(\underline{\boldsymbol{x}}^{*(k)})$

is replaced by: $\Theta_j^{(k+1)} = \max \{ 0, \Theta_j^{(k)} + \varphi_j(\underline{x}^{*(k)}) \} , \quad j=1, 2,\dots,m$

  - **Version developed by Wierzbicki** (1971) - improves behaviour of the algorithm in ‘peculiar’ situations

Penalty function with 'shifting' the penalty term – geometric interpretacion



Numerical methods of optimization – IAEA Fellowship, Birmingham University, 1969/70

- Analysing and selecting most suitable algorithms
- Coding and testing the selected algorithms (FORTRAN, emc KDF9 and IBM 1130);
- Comparison of methods and computer codes
  - Constrained problem – method of Powell's penalty function (with 'shifting' of the penalty term) (Powell, 1968); use of '*slack variables*'
  - Four methods used for unconstrained minimization:
    - Rosenbrock, 1960 – '*Orthogonal directions*' (code **COROS)**
    - Davies, Swan, Campey, 1965 ; (code **DISCON)**
    - Powell, 1968 – '*Conjugate directions*'; (code **PCON)**
    - Stewart, 1967 '*Variable metric method*' (code **CONVAR)**

Selected methods for unconstrained minimization - overview

- **Rosenbrock's method** - attempt to find the direction of ridge and considering it as 'promissing' search direction
  - Starting point: $\underline{x}_o$; $\Phi_o = \Phi(x_o)$
  - n orthogonal directions: $\xi_1, \xi_2, \ldots \xi_n$
  - Series of searches along these directions (with given step e_i):
    - $\underline{x}_i = \underline{x}_i + e_i \, \underline{\xi}_1$, $\Phi = \Phi(\underline{x})$
    - if $\Phi \leq \Phi_o$ 'successfull step' $\Phi_o = \Phi$; $e_i = \alpha \, e_i$; $(\alpha > 1)$
    - if $\Phi > \Phi_o$ 'unsuccessfull step' $e_i = - \beta \, e_i$; $(0 < \beta < 1)$

    This search is continued in each direction ξ_i in turn until at least one trial is 'successful' in each direction, and one has failed
  - ξ_1 is replaced by $\xi^*_1 = \Sigma_i^n d_i \, \xi_i$, where d_i (i=1, 2,…,n) – the algebraic sum of all sucessive steps e_i in the direction ξ_i
  - The remaining new search directions are obtained by orthogonalization process, and the iterative process is repeated
- **DSC (Davis, Swan, Campey)** method - analogical to Rosenbrock's method, but using unidimensional minimization along the directions

Selected methods for unconstrained minimization - overview (cont.)

❑ Conjugate directions method of Powell (1964)

- If the directions $\xi_1, \xi_2, \ldots, \xi_n$ are mutually conjugate with respect to the positive definite (psd) quadratic objective function

$$\Phi(\underline{x}) = c + \underline{b}^T \underline{x} + \tfrac{1}{2} \underline{x}^T G \underline{x}$$

than the minimum of this function can be found by minimizing the function along these directions n times

Comment: Directions $\underline{p}$ i $\underline{q}$ are defined to be conjugate with respect to the psd quadratic objective function, if they are both non-zero, and if they satisfy the condition:

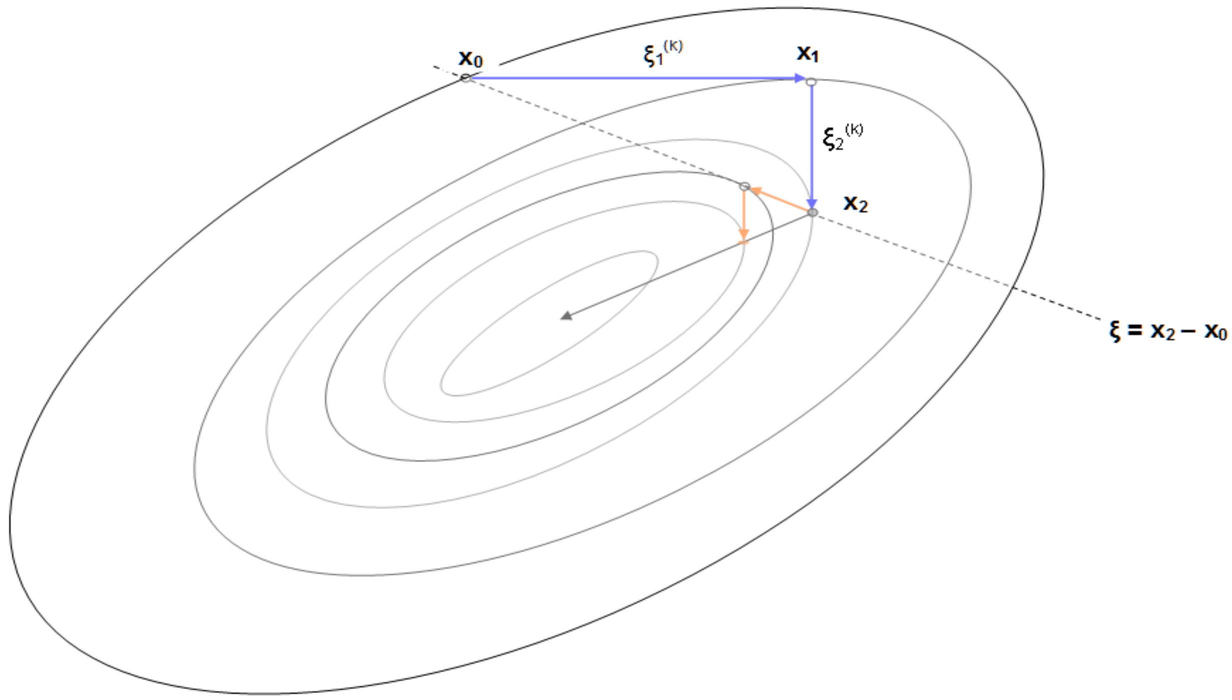
$$\underline{p}^T G \underline{q} = 0$$

Selected methods for unconstrained minimization - overview - cont.

- **Conjugate direction method** (Powell, 1968),
  - **Powell's iteration** requires:
    - n independent search directions $\xi_1, \xi_2,\ldots,\xi_n$ (initially the coordinate directions) and starting point x_0 ;
    - „n" minimizations along each of the search direction in turn, changing the estimate $x_o \rightarrow x$, $\Phi(\underline{x}) < \Phi(\underline{x}_0)$
    - **new search direction** $\xi = x - x_o$ replaces one of the base directions in each iteration (version I of algorithm),
  - If function $\Phi(\underline{x})$ is (psd) quadratic function, then n iterations will find the minimum; the **search directions become mutually conjugate**.
  - In version II (1968) special measures are taken to retain linear independece (therefore, more than n iteration may be required)

Conjugate direction method (Powell 1964)

Example explaining how the conjugate directions are generated



Selected methods for unconstrained minimization - Stewart's method

Algorithm of Stewart (1967) – an extension of DFP (Davidon, Fletcher, Powell, 1963) – originally called „the variable metric" method is also a conjugate direction method

The k[th] iteration of Davidon's method changes the estimate $\boldsymbol{x}_k$ to the estimate $\boldsymbol{x}_{k+1}$ by searching for the minimum of the objective function along the direction defined by vector

$$\underline{d}_k = - H_k \underline{g}_k$$

where: $H_k \in R^{nxn}$ equivalent of the matrix G^{-1} used in Newton's method (the matrix of second derivatives $[\nabla^2 \Phi(x_k)]^{-1}$,

$\underline{g}_k$ – gradient $\nabla\Phi(\boldsymbol{x}_k)$

Vector of variables is updated according to the formula:

$$\mathrm{x}_{k+1} = \mathrm{x}_k - \alpha_k H_k \mathrm{g}_k, \qquad a_k = \arg\min_{a_k > 0} \Phi(\underline{x}_k + a_k \underline{d}_k)$$

Selected methods for unconstrained minimization - Stevart's method, (cont.)

Matrix *H* is updated in each iteration using the formula:

$$H_{k+1} = H_k - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k} + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k}. \qquad \text{VM}_{\text{DFP}}$$

where:

$$\delta_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k,$$

Fletcher & Powell provided theoretical bases of this algorithm:

- If the matrix H_1 is initially chosen to be psd, then this property is retained by the subsequent matrices
- If $\Phi(x)$ is a psd quadratic form, then the iteration terminates after at most n iterations (quadratic convergence property) and the matrix H_{n+1} is the inverse of the second derivative matrix of the objective function $\Phi(x)$ ($H_{n+1} = G^{-1}$)

Selected methods for unconstrained minimization - Stevart's method, (cont.)

- **In Stewart's** algorithm the first derivatives are approximated by finite differences:

$$\partial\Phi/\partial \boldsymbol{x}_i = [\Phi(\underline{\mathbf{x}}+\mathrm{h}_i\,\underline{e}_i) - \Phi(\underline{\mathbf{x}})]/\mathrm{h}_i\,, \qquad \mathrm{i} = 1, 2, \ldots, \mathrm{n}$$

where: h_i approximation step, and e_i is the unit vector, with its i-th component of unity and its other components 0.

The formula of central differences is used at certain situations:

$$\partial\Phi/\partial \boldsymbol{x}_i = [\Phi(\underline{\mathbf{x}}+\mathrm{h}_i\,\underline{e}_i) - \Phi(\underline{\mathbf{x}}-\mathrm{h}_i\,\underline{e}_i)]/(2\mathrm{h}_i), \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n}$$

Approximation step h_i is calculated with a view to **balancing truncation and round-off errors**; taking into account the curvature of Φ along the direction e_i, given as diagonal elements of matrix $\Lambda = H_{k+1}^{-1}$, (calculated using a recurence formula depending on g_k, γ_k, H_k)

Numerical methods for constrained optimization
IAEA Fellowship, Birmingham University, 1969/70

- Analysis of selected methods and development of the computer codes
- Comparison of methods; limited number of testing problems

Problem designation	Author	Characteristic feature		Number of variables (n)	Number of constraints	
		Objective f	Constraints		Total (m)	Inequalities (m_1)
T1	MK 1970	NL	KW	2	2	2
T2	MK 1970	KW	NL	3	3	3
T3	MK 1970	SNL	NL	12	3	3
HEX	MK 1970	SNL	SNL	3	3	3

*) NL – nonlinear, SNL – strongly nonlinear, KW - quadratic

Numerical methods for constrained optimization - IAEA Fellowship, Birmingham University, 1969/70

Program name	Problem T1		Problem T2		Problem T3		Problem HEX	
	N_i*	N_f**	N_i*	N_f**	N_i*	N_f**	N_i*	N_f**
COROS	6	1019	5	1758	6	9787	5	3138
DISCON	6	986	4	1233	6	6246	2	2134
PCON	6	580	4	776	6	7013	3	1798
CONVAR	6	359	4	381	6	1925	2	768

*) Number of iterations (unconstrained minimizations) UnOpt

**) Total number of calculations of the model (i.e. objective function and constraints)

Conclusions

- Effectiveness of the selected method of dealing with constraints confirmed; (*'Penalty function with shifting'*, Powell, 1969);
- Unconstrained minimization methods based on '*variable metric*' formulas; (Stewart,1967 /CONVAR code) seemed to be very promising

Numerical methods for constrained optimization - IEA Świerk, 1971 - 79

- **MINCON code** developed in IAE in 1970-s uses the algorithm of Wierzbicki (1971), modified to handle equality and inequality constraints (based on Powell, 1968; Michalski et al, 1970)

$$\Phi(x,\sigma,\Theta) = F(x) + P(\sigma_j,\Theta_j,\varphi_j,\psi_j)\ ;$$

$$P(\sigma_j,\Theta_j,\varphi_j,\psi_j) = \sum_{J_1} \sigma_j\,[\varphi_j + \Theta_j]\, max[0,\ \varphi_j + \Theta_j] + \sum_{J_2} \sigma_j[\psi_j + \Theta_j]^2$$

where: J_1 set of indices j for inequality constraints $\varphi_j(\underline{\mathbf{x}}) \leq 0$

J_2 set of indices j for equality constraints $\psi_j(\underline{\mathbf{x}}) = 0$

Modification of penalty parameter Θ based on the formula:

$$\Theta_j^{(k+1)} = \max\,[\,0,\ \varphi_j(\underline{x}^{*(k)}) + \Theta_j^{(k)}\,] \qquad j \in J_1$$

$$\Theta_j^{(k+1)} = \psi_j(\underline{x}^{*(k)}) + \Theta_j^{(k)} \qquad j \in J_2$$

Constrained optimization method (MINCON Code)

- **MINCON code - Constrained Optimization (Wierzbicki,1971)**
- Four situation depending on the degree of exceeding the constraints:
- *'Unacceptable'* and *'Acceptable'*

Defined in terms of two sets:

$G_d = \{ \boldsymbol{x} \in R^n\text{:}\ \ h_j < d^{(k)}\ ;\ j=1,2,\ldots,\ m \}$ unaccceptable

$G_c = \{ \boldsymbol{x} \in R^n\text{:}\ \ h_j < c^{(k)}\ ;\ j=1,2,\ldots,\ m \}$ acceptable

where: $d^{(k)} > c^{(k)} > 0$

$h_j = f_j(\underline{\mathbf{x}}^{(k)})\ ;$ $j \in J_1$ inequality constraints

$h_j = abs\,[f_j(\underline{\mathbf{x}}^{(k)})]\ ;$ $j \in J_2$ equality constraints

Constrained optimization method (MINCON Code)

START

$\underline{\boldsymbol{\sigma}} = \underline{\boldsymbol{\sigma}}_s$, $\quad \underline{\boldsymbol{\Theta}} = 0$, $\quad d = r_D \varepsilon_y$, $\quad c = r_A d$

NI = 0, $\underline{\mathbf{x}}_0 = \underline{\mathbf{x}}_s$

A

NI = NI +1, $\quad \underline{\mathbf{x}}_0 = \underline{\mathbf{x}}^*$

Minimize the function $\Phi(\underline{\mathbf{x}}, \underline{\boldsymbol{\sigma}}, \underline{\boldsymbol{\Theta}})$ with regard to $\underline{\mathbf{x}}$ (Unc. Opt. probl.)

$$\Phi(\underline{\mathbf{x}}, \underline{\boldsymbol{\sigma}}, \underline{\boldsymbol{\Theta}}) = F(\underline{\mathbf{x}}) + \sum_{j \in J_1} \sigma_j [\, f_j(\underline{\mathbf{x}}) + \Theta_j \,] \max[\, 0, f_j(\underline{\mathbf{x}}) + \Theta_j \,] + \sum_{j \in J_2} \sigma_j [\, f_j(\underline{\mathbf{x}}) + \Theta_j \,]^2$$

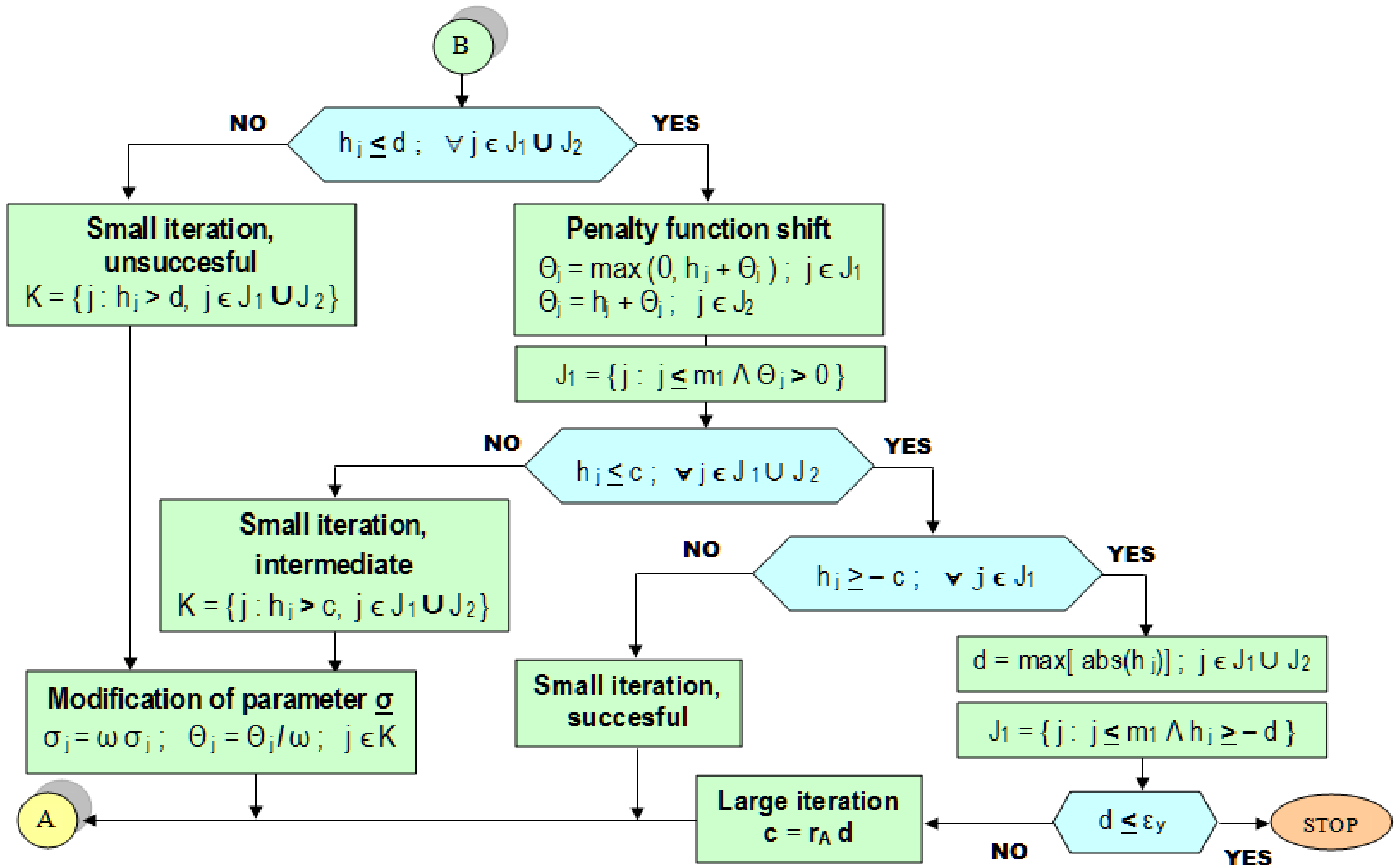
$J_1 = \{ 1, 2, \ldots, m_1 \}$; $\quad J_2 = \{ 1 + m_1, 2 + m_1, \ldots, m \}$

Starting point: $\underline{\mathbf{x}} = \underline{\mathbf{x}}_0$; $\quad$ Solution: $\underline{\mathbf{x}}^*$

$h_j = f_j(\underline{\mathbf{x}}^*)$, $\; j \in J_1$; $\quad h_j = \text{abs}\,[\, f_j(\underline{\mathbf{x}}^*) \,]$, $\; j \in J_2$

B

Constrained optimization method (MINCON Code)



enco

Effectiveness of the selected codes for unconstrained minimization (Himmelblau, 1971)

Total number of calculations of the function needed for the solution (cited by Himmelblau, 1971)

Problem	Author	Charact. of func. / Number of variables	Optimization algorithm (UncOpt)						
			Hook, Jeeves	Nelder, Mead	Rosen-brock	Powell, 1964		Stewart, 1967	
						QI	GR	QI	GR
HM-1	Zangwill,1967	Polynomial 2-dg./ 2	80	185	62	29	218	16	84
HM-2	White,Holst,1964	Polynomial 6-dg./ 2	651	359	294	284	156	256	194
HM-3	–	Polynomial 3-dg./ 2	640	190	163	24	220	a	a
HM-4	Beale,1958	Polynomial 6-dg./ 2	205	230	218	134	396	f	161
HM-5	Engwall,1966	Polynomial 4-dg./ 2	64	210	119	96	264	119	137
HM-6	Box, 1966	Sum of sq exp./ 2	498	268	314	161	278	177	406
HM-7	Zangwill,1967	Polynomial 2-dg./ 3	130	810	297	84	502	37	108
HM-9	Engwall,1966	Sum of sq nl terms./3	81	561	457	315	652	150	304
HM-10	Fletcher,1963	Strongly nonlinear/ 3	1230	566	513	48	277	191	430
HM-11	Bard,1970	Sum of sq nl terms./3	–	711	–	102	174	134	198
HM-12	Powell,1964	Polynomial 4-dg./ 4	77	1022	801	966	1783	622	1117
HM-13	Crag, Levy,1969	Strongly nonlinear / 4	9283	563	955	3480	3103	1662	3749
HM-14	Wood	Polynomial 4-dg./ 4	836	797	1043	276	850	715	905

QI - Linear minimization along the search direction - using quadratic interpolation
GR - Linear minimization along the search direction using 'golden ratio"
a - converge to global solution (– ∞); **f** - solution not found

Numerical methods for Unconstrained Optimization — Comments/conclusions with regard to the effectiveness

- **UncOpt methods tested in 1970 r. (IAEA Fellowship)**
- **Repeating linear minimizations**; process ‘expensive’ with regard to the effectiveness (in terms of function evaluations):
  - **PCON (Powell) & DISCON (Davis, Swan, Campey) —** n+1 or n times in each step (iteration)
  - **CONVAR** (Davidon, Stewart) — once in each iteration
- **CONVAR requires numerical approximation of gradient** of the function Φ - (n+1 additional evaluations of Φ, or more, if central differences are needed)
- **Approximation of gradient** of Φ may be made less laborious, when the algorithm used for calculating the model (functions Φ(x), φ(x), ψ(x) has a block structure.

Numerical methods of Unc Opt — comments / conclusions with regard to the effectiveness

- **Favorable feature** of CONVAR code (or other methods based on 'variable metric' approach) is **possibility of using the second order information accumulated in the** subsequent iterations of ConOpt (transferred in the form of matrix H); this has a positive impact on the effectiveness :

$$G_{k+1} \approx G_k + 2\sum_{j=1}^{m} \triangle\sigma_j \nabla f_j \nabla f_j^T$$

- Calculating the matrix $H_{k+1} = (G_{k+1})^{-1}$ does not require **inverting the matrix G_{k+1}**; it is calculated recursively by application of the following formula:

$$(G_{k+1})^{-1} = G_k^{-1} - \frac{G_k^{-1} \nabla f_j \nabla f_j^T G_k^{-1}}{(2\triangle\sigma_j)^{-1} + \nabla f_j^T G_k^{-1} \nabla f_j}$$

for $G_k^{-1} = H_k$

Numerical methods of Unc Opt –
increasing the effectiveness of the calculation

- **Use of algorithms that eliminate the need for linear searches, e.g.** those based on the rank one formula:

$$H^* = H + \frac{(\delta - H\gamma)(\delta - H\gamma)^T}{\gamma^T(\delta - H\gamma)} \qquad \text{VM (1)}$$

- **Unpleasant side effects of this formula:**
  - The matrices H **no longer remain psd**
  - The correction term **may happen to be unbounded**
  - The matrix **may become singular or undetermined**
- Thus, the use of this correction **requires that sometimes a special strategy is needed**, which makes the routine operations more complicated

Numerical methods of Unc Opt –
increasing the effectiveness of the calculation

- **Davidon's new method** (Version II, 1968)

$$\delta = -H\,g$$

$$H^* = H + (1-\lambda)\frac{(\delta - H\gamma)(\delta - H\gamma)^T}{(\delta - H\gamma)^T(g+\gamma)} \qquad \text{VM (2)}$$

- Matrix H* is **kept psd** by a suitable choice of λ; its value depends on the parameter:

$$\rho = -\frac{(\delta - H\gamma)^T\,g}{(\delta - H\gamma)^T\,(g+\gamma)}$$

- Typically the parameter *λ* is such that the formuła VM (2) remains identical to the rank one formula VM (1); however, in certain conditions *λ* is different, and its choice is intended to **prevent unwarranted extrapolation about the behaviour of the function *Φ***

Numerical methods of Unc Opt – increasing the effectiveness of the calculation

- **Version II of Davidon's method (1968), cont.**

- **(a)** $\lambda = \alpha$ **if** $-\alpha \dfrac{\alpha}{1+\alpha} \leqslant \rho < \dfrac{\alpha}{1-\alpha}$
- **(b)** $\lambda = -\dfrac{\rho}{\rho+1}$ **if** $-\dfrac{\beta}{1+\beta} \leqslant \rho < -\dfrac{\alpha}{1+\alpha}$
- **(c)** $\lambda = \beta$ **if** $-\dfrac{\beta}{\beta-1} \leqslant \rho < -\dfrac{\beta}{\beta+1}$
- **(d)** $\lambda = \dfrac{\rho}{\rho+1}$ **otherwise, (in this case formulas VM1 i VM2 become identical)**
- **The algorithm terminates when:**

$$\Delta\Phi_E = \frac{1}{2}(\delta - H\gamma)^T(g+\gamma) \leq \frac{1}{2}\varepsilon$$

Numerical methods of Unc Opt – increasing the effectiveness of the calculation

Fletcher's algorithm (1970) – other, new formula for correcting the matrix H:

$$H^* = (I - \frac{\delta\,\gamma^T}{\delta^T\gamma})\,H\,(I - \frac{\gamma\,\delta^T}{\delta^T\gamma}) + \frac{\delta\,\delta^T}{\delta^T\gamma} \qquad VM_{FL}$$

The matrix defined by the formula VM_{FL} is 'less singular' as compared to the original DFP (formuła VM_{DFP}), but may cause H^* to tend to become unbounded. For this reason Fletcher proposed to use the convex combination of the formulas VM_{DFP} i VM_{FL}:

$$H^*_\varphi = (1-\varphi)\,H^*_{DFP} + \varphi\,H^*_{FL})$$

Where: $0 \leq \varphi \leq 1$

$\varphi = 0$ if $\delta^T\gamma < \gamma^T H\,\gamma$

$\varphi = 1$ if $\delta^T\gamma \geq \gamma^T H\,\gamma$

Fletcher's algorithm (1970), cont.

Abandonment of linear searches requires some means to retain a sufficiently large decrease of Φ in each iteration.

$a_k = 1$ only if the condition $$\mu \leq \frac{\Phi(x) - \Phi(x - \alpha H g)}{g^T \delta} \leq 1 - \mu$$

is satisfied, where $0 < \mu \ll 1$

If the left hand side inequality is violated than a_k is calculated by cubic interpolation. If a_k is too small and the right hand side inequality is violated then $\alpha_k = 1/w, 1/w^2, \ldots$, where $0 < w < 1$

In addition, the value of a_k needs to satisfy the condition $\delta^T y > 0$, if this condition is not satisfied then the values $\alpha_k/w, \alpha_k/w^2, \ldots$, are tried.

The algorithm is terminated when: $| \delta | = | -Hg | \leq \varepsilon_x$

Numerical methods of optimization - improving effectiveness and reliability of the calculation

Termination criteria - important element of the algorithm

(1) $\Phi(x) - \Phi(x^*) \leq \varepsilon_\Phi$

(2a) $\dfrac{|\, x_i - x_i^* \,|}{1 + |\, x_i \,|} \leq \varepsilon_i$

(2b) $max\left(\dfrac{|\, x_i - x_i^* \,|}{1 + |\, x_i \,|}\right) \leq \varepsilon_x \; ; \qquad i = 1, 2, ..., n$

(3) $max\, [\, g_i(x) - g_i(x^*) \,] \leq \varepsilon_G$

x* , *x* – Values of variables in two subsequent iterations

ε_Φ , ε_i , ε_x , ε_G – constant parameters

Numerical methods of optimization - improving effectiveness and reliability of the calculation

Termination criteria (constrained optimization algorithm)

Making the parameters ε, ε_Φ dependent on the difference between the function value F found in the iteration k and the estimated value of the function at the solution point $\hat{x}$

$$\Delta F_k = F(\hat{x}^{(k)}) - F(\hat{x}) \;; \qquad \varepsilon^{(k+1)} = \delta\, \Delta F_k \,; \qquad 0 < \delta \ll 1$$

$$\Delta F_k \approx E_k = -2 \sum_J \sigma_j^{(k)} [\Theta_j^{(k)} + f_j(\hat{x}^{(k)})]\, f_j(\hat{x}^{(k)})$$

J – The set of indices corresponding to the active constraints

Another useful criterion: $|\, E_k(x) - E_k(x^*) \,| \,/\, |\, E_k(x) \,| \le \delta_E$

$\underline{x}$, $\underline{x}^*$ - Values of variables in two subsequent iterations of unconstrained optimization

Development of MINCON code (cont.)

Selecting algorithm for unconstrained optimization

Testing 3 algorithms for Unc Opt considered most promising

- ❑ Davidon's formula 1968 (VM2) – **MIDAS** code
- ❑ Fletcher's formula 1970 (VM_{FL}) – **MIFLE** code
- ❑ Coniugate Gradients of Powell 1968 – **MINPO** code

- ➢ MIDAS & MIFLE differ in some details as compared to the original routines (e.g. definition of starting matrix H, redefinition of H in 'abnormal' situations, termination criteria for UncOpt, etc.)
- ➢ Testing problems (13) according to Himmelblau, 1971; comparison of the effectiveness using the same criteria (CYBER 70, single precision)
- ➢ Numerical experiments included the investigation of effects of changing the values of parameters used in these algorithms

Results of numerical tests – comparison of unconstrained optimization codes

Problem	Author	Objective function/ Number of variables	Himmelblau,1971		Kulig,1980		
			Powell	Stewart	MINPO Powell,1964	MIDAS Davidon	MIFLE Fletcher
HM-1	Zangwill,1967	Polynomial 2-dg./ 2	29	16	30 - 35	9 - 15	6
HM-2	White, Holst,1964	Polynomial 6-dg./ 2	284	256	177 - 228	108 - 238	163
HM-3	–	Polynomial 3-dg./ 2	24	a	38 - 42	19 - 48	35
HM-4	Beale,1958	Polynomial 6-dg./ 2	134	f	118 - 123	44 - 152	60
HM-5	Engwall,1966	Polynomial 4-dg./ 2	96	119	72 - 73	41 - 68	48
HM-6	Box, 1966	Sum of sq exp. terms/ 2	161	177	87 - 93	64 - 153	119
HM-7	Zangwill,1967	Polynomial 2-dg./ 3	84	37	78 - 79	96 - 144	37
HM-9	Engwall,1966	Sum of sqs nl. terms./3	315	150	201 - 225	109 - 228	155
HM-10	Fletcher,1963	Strongly nonlinear/ 3	48	191	188 - 190	154 - 287	220
HM-11	Bard,1970	Sum of sqs. nl. terms./3	102	134	142 - 190	93 - 115	166
HM-12	Powell,1964	Polynomial 4-dg./ 4	966	622	414 - 517	212 - 314*	284*
HM-13	Crag,Levy,1969	Strongly nonlinear / 4	3480	1662	524 - 848	216 - 500*	674*
HM-14	Wood	Polynomial 4-dg./ 4	276	715	688 - 951	172 - 307	725

a - Converge to global solution (– ∞); **f** - Solution not found

*) Accuracy of the solution $\Delta x \sim 10^{-3}$ too low ($\Delta x \sim 10^{-4}$ – increases the number of function evaluations ~50%)

Results of numerical tests – comparison of constrained optimization codes (MINCON & MIPOW)

Probl.	Author	Probl. features Obj. f./ constr.	n	m_1/m_2	MIPOW	MINCON	Other codes	Comments
K1	Rosenbrock,1960	NL / L	3	8/ 0	307/2	250/2	310 - 576	Box, COMPLEX; ε_x~E-7
K2	Box, 1965	SNL / L	2	5/ 0	122/4	75/4	159	Box, COMPLEX; ε_x~E-6
K3	Kulig, 1970	NL / L	2	2/ 0	228/5	90/5	580/ 359	PCON/CONVAR
K4	Kulig, 1970	KW / L	3	3/ 0	261/5	157/5	776/ 381	PCON/CONVAR
K5	Szymanowski,1972	NL / NL	2	3/ 0	261/4	137/4		
K6	Szymanowski,1972	NL / NL	2	5/ 0	118/3	68/3		
K7	Szymanowski,1972	NL / NL	2	6/ 0	282/7	163/7		
K8	Powell, 1968	SNL / NL	5	0/ 3	493/3	294/3		
K9	Box, 1965	NL / KW	5	15/ 0	828/5	556/5	1440	Box, COMPLEX; ε_x~E-4
K10	Kulig, 1972	SNL / NL	20	2/ 4	12220/8	1049/8		
K11	Kulig, 1972	KW / KW	6	6/ 2	955/6	458/8		
K12	Kulig, 1972	NL / KW	40	2/ 0	4635/6	2288/6		
EWA	Kulig, 1980	SNL / NL	6	11/ 0	1371/4	630/5		
TURB	Kulig, 1979	SNL/SNL	16	0/ 11	7509/7	7249/7		

L – linear function, KW – quadratic function, NL – nonlinear function, SNL – function strongly nonlinear

IEA work - Conclusions / discussion

- Numerical tests confirmed the **effectiveness of the penalty function** method (with penalty ‘shift’ by Wierzbicki, 1971); consideration of active constraints
- **Converting inequality constraints into equations** using the ‘slack variables’ should be used with care (effectiveness, reliability ?)
- **‘Variable Metric’** (VM) methods of UncOpt more effective as compared to the conjugate gradient methods
  - Potential **‘savings’ in the approximation of gradients (when algorithm of** the **model** has a **block structure**)
  - Possibility of easy **transfer of second order information** on the function (in the form of Hessian) from one constrained iteration into the next

IEA work - Conclusions / discussion, (cont.)

- Algorithms based on VM formulas are more sensitive for **the effects of truncation/ round-off**, and **incorect estimation of Hessian matrix** (both in starting and in subsequent updates)
- **Gradient approximation - arbitrary definition of intervals** more reliable than automatic selection proposed by Stewart.
- **Termination criteria for Unc and Con Opt** – important element of the algorithm; **parameters related to 'accuracy' should** depend on the iteration, i.e., based on the estimated difference of the function $\Delta F_k = F(x_k) - F(\hat{x})$

Selected References

- Rosenbrock, H.H., „An automatic method for finding the gratest or least value of a function”, The Computer Journal, Vol. 3, 1960.
- Powell, M.J.D. – An iterative method for finding stationary values of a function of several variables, The Computer Journal, Vol. 5, 1962.
- Powell, M.J.D., „An efficient method for finding the minimum of a function of several variables without calculating derivatives”, The Computer Journal, Vol. 7, 1964, 155-162.
- Stewart, G.W. – A modification of Dawidon's minimization method to accept difference approximation of derivatives, J. Assoc. Comp. Mach. Vol. 14, 1967.
- Davidon, W.C. – Variance algorithm for minimization, The Computer Journal, Vol. 10, 1968.
- Powell, M.J.D., „A method for non-linear constraints in minimization problems”, Proceedings of BCS/IMA Conference n/t „Optimization”, Keele University, 1968.
- Fletcher, R., A new approach to variable metric algorithms. The Computer Journal, Vol. 13, 1970.
- Wierzbicki, A., „A penalty function shifting method in constrained optimization and its convergence properties” Arch. Aut. i Tel., XVI-4, 1971

Part 3

Numerical optimization methods -
Achievements in the field

What has changed in the field of NLP since 1970s ?

What has changed in the field of NLP since 1970s ?

- Well established **infrastructure** for the development and practical implementation of algorithms
  - Enhanced capabilities of **computer hardware**
  - **Software libraries**
  - **Benchmarking** guidelines and tests
- Significant **achievements in numerical analysis/ mathematical programming** — new theoretical discoveries / ideas, convergence analysis, etc.
- Wide selection and availability of NLP **optimization algorithms**
- Increased a role of mathematicians and mathematical software specialists in the development of software (?)

enco

Existing Software Libraries

➢Several recognized software libraries:

- **IMSL**, Rogue Wave Software Library, Inc. (USA, UK, Germany, France, Japan), www.vni.com
- **NAG** High Performance Computing services,

  https://www.nag.co.uk/content/high-performance-computing-consulting-and-services
- **NLopt** - a free/open-source library for NLP started by S. G. Johnson, https://nlopt.readthedocs.io/en/latest/
- AERE, Harwell Subroutine Library, England
- Universities, UK – Cambridge (DAMTP), Dundee
- USA – Princeton, Stanford

Benchmarking of algorithms

- **Well established standards and guidelines**
  - reasons for benchmarking, - selecting the test set, - performing the experiments, - analysing and reporting the results, etc.

- **Availability of appropriate benchmarking tests**
  - Cornwell, L.W., et al. „Test Problems for Constrained mathematical programming algorithms”, ANL-AMD-TM-320, July 1978. https://www.osti.gov/servlets/purl/6576051
  - K. Schittkowski, „306 Test Problems for Nonlinear Programming with Optimal Solutions, User Guide”, 2009. http://www.klaus-schittkowski.de/tpnp.htm
  - The CUTEr/st Test Problem Set, http://www.cuter.rl.ac.uk/Problems/mastsif.shtml

enco

Methodological basis - relevant aspects/issues

- **Lagrangian functions / Lagrangian multipliers** become an important analytical tool/basis for the optimality conditions
- A role of **penalty function methods** diminished (?)
- **Linear/quadratic approximation** plays important role in converting the original NLP into a sequence of problems simpler/easier to solve
- Unconstrained minimization algorithms still play an important role in solving general NLPs
- Increased application of the concept of „trust region” and the **trust-region method**
- Use of additional safeguards to **stabilize algorithms** for increased reliability and the rate of convergence (line search, trust region, a filter concept, factorization of relevant matrices, etc.)

Lagrangian Function/Lagrangian multipliers

➢Normal Lagrangian function

Problem: $\underset{x \in X_0}{minimize}\, f(x); \qquad X_0 = \{x \in \mathbb{R}^n : \; g(x) \leqslant 0 \in \mathbb{R}^m\}$ (1)

Where f : $R^n \to R^1$ and g : $R^n \to R^m$ - twice continuously differentiable

$$L(y,x) = f(x) + y^T g = f(x) + \sum_{i \in I} y_i\, g_i(x) \qquad I = \{1, \ldots, m\} \qquad (2)$$

If the problem is normal and functions f(x) and g(x) are convex, then the necessary and sufficient conditions for optimality is that exist a vector of Lagrangian multipliers $y^* \in \mathbb{R}^m_+$ such that L(y,x) has at y*,x* its global saddle point

$$\underset{y^* \in \mathbb{R}^m_+}{L(y,x^*) \leqslant L(y^*,x^*)} \leqslant \underset{x^* \in \mathbb{R}^n}{L(y^*,x)} \qquad (3)$$

The relation: $\underset{x \in \mathbb{R}^n}{min}\; \underset{y \in \mathbb{R}^m}{max}\; L(y,x) = L(y^*,x^*) = f(x^*)$

valid for convex problem only

Lagrangial Function/Lagrangian multipliers (cont.)

Necessary condition for optimality in differential problem (KKT)

$$\nabla_x L(y^*, x^*) = \nabla_x f(x^*) + y^{*T} \nabla_x g(x^*) = \nabla_x f(x^*) + \sum_{i \in I} y_i^* \nabla_x g_i(x^*) = 0 \in \mathbb{R}^n \qquad (4)$$

and

$$g(x^*) \leqslant 0; \quad y^{*T} g(x^*) = 0; \quad y^* \geqslant 0 \ \in \mathbb{R}^m \qquad (5)$$

The KKT conditions (4) and (5) are only necessary for optimality. To become sufficient, they must be supplemented by a **second order condition**:

$$x^T H(y^*, x^*)\, x > 0 \qquad for\ all\ x \neq 0 \ \ such\ \ that\ \ A_{SA} x = 0$$

Where:

H is the **Hessian matrix** of $L(y^*,x^*)$:

$$H(y^*, x^*) = \nabla_x^2 L(y^*, x^*) = \nabla_x^2 f(x^*) + \sum_{i \in SA} y_i^* \nabla_x^2 g_i(x^*)$$

A_{SA} is the **Jacobian matrix** for strongly active constraints

$$A_{SA} = \nabla_x g^s(x^*) = \nabla_x g_i(x^*) \in \mathbb{R}^{s,n},\ i \in SA,\ \ where\ \ SA = \{i \in I :\ g_i(x) = 0,\ y^* > 0\}$$

H**(y*,x*) should be psd in the subspace of directions tangent to strongly active (SA) constraints**

Wierzbicki, A., „A Quadratic Approximation Method Based on Augmented Lagrangian Functions for Nonconvex Nonlinear Programming Problems, IIASA, Laxenburg, Austria (1978)

Lagrangial Function/Lagrangian multipliers (cont.)

- **Equality and inequality constraints**

  $\underset{x^* \in \mathbb{R}^n}{min} f(x) \quad h_k(x) = 0,\ k = 1, \ldots, m_e \quad g_k(x) \leqslant 0;\ k = 1, \ldots, m_i$

- **Lagrange function:** $L(x, \lambda, \mu) = f(x) + \sum_{k=1}^{m_E} \lambda_k\, h_k(x) + \sum_{k=1}^{m_I} \mu_k\, g_k(x)$

- **Karush-Kuhn-Tucker (KKT) optimality conditions:**

  (1) Stationarity conditions.... $\nabla f(x) + \sum_{k=1}^{m_e} \lambda_k \nabla h_k(x) + \sum_{k=1}^{m_i} \mu_k \nabla g_k(x) = 0$

  (2) Primal feasibility conditions................ $h(x) = 0$, $g(x) \leq 0$

  (3) Dual feasibility conditions..................... $\forall k \in \{1, \ldots, m_i\},\ \mu_k \geq 0$

  (4) Complementarity conditions....... $\forall k \in \{1, \ldots, m_i\},\ \mu_k g_k(x) = 0$

Augmented Lagrangian Function

Augmented Lagrangian function

$$L(\underline{x}, \lambda, r) = f(\underline{x}) - \sum_{i=1}^{m_E} \lambda_i h_i(\underline{x}) + r \sum_{i=1}^{m_E} [h_i(\underline{x})]^2 , \qquad (1)$$

- Proposed by Fletcher (1975) for solving constrained problem with equality constraints: $h_i(\underline{x}) = 0, \ \ i=1,\ldots m_E$ (2)

  The method performs unconstrained optimization to calculate the stationary point $\underline{\lambda} = \underline{\lambda}^*$, $\underline{x} = \underline{x}^*$ of (normal) Lagrangian function (3)

$$f(\underline{x}) - \sum_{i=1}^{m_E} \lambda_i^* h_i(\underline{x}) \qquad (3)$$

- Increasing the value of r usually forces the stationary point to be a minimum
- The calculated vector $\underline{x}$ that minimize the function (1) for several values of the parameters $\underline{\lambda}$ and r is then used to adjust $\underline{\lambda}$ so that they converge to $\underline{\lambda}^*$,
- Hence $\underline{x}(\underline{\lambda}, r)$ converges to the required solution $\underline{x}^*$ of the original constrained problem (with a moderate value of the parameter r)

Nonlinear optimization algorithms - overview

- **Two main approaches for solving constrained NLP’s:**
- **Sequential quadratic programming (SQP)**

  Example solvers in NAG Library – e04vh, e04uc, e04us;
  - Based on Gill, P.E., et al. (2005) „SNOP: An Algorithm for Large-scale Constrained Optimization, SIAM Review 47(1), 99-131.
- **Interior point method (IPM)**
  - Based on Wächter, A., Biegler, L.T., (2006) „On the implementation of a primal- dual interior point filter line search algorithm for large-scale nonlinear programming”, Mathematical Programming 106(1), 25-57.
  - Example solver in NAG Library – e04st

Nonlinear optimization algorithms - overview

Distinct features of SQP and IPM methods:

- how the inequality constraints are treated
- how the solver approaches the optimal solution (the progres of the optimality measures: optimality, feasibility, complementarity)
- 'Twofold nature' of the inequality constraints:

  If the optimal point satisfy strictly the inequality, it is **non-active** (could be removed from the model); if it is satisfied as an equality (i.e. **active at the solution**),the constraint should be present from the very beginning;

Sequential Quadratic Programming (SQP) - overview

- Most of the existing **SQP solvers** based on **active set approach**
- At each iteration it solves a quadratic approximation of the original problem

Initialization:
- choosing a first estimate of the solution $\underline{x}_0$;
- building a quadratic model of the objective around $\underline{x}_0$;
- taking a first guess of the set of active constraints;

Iteration k:
- solving the quadratic program by active set estimation;
- updating $\underline{x}_{k+1}$ and the set of active constraints;
- building a new quadratic model around $\underline{x}_{k+1}$;

Sequential Quadratic Programming (SQP) - overview

➢ Main characteristics of SQP methods

- Perform lots of inexpensive iterations
- Work on the 'null space' of the constraints / 'walk along the boundary' of the feasible region determined by the constraints. The iterates are thus early on feasible with regard to all linear constraints (and local linearization of nonlinear constraints)
- The more active constraints there are, the cheaper the iterations are. As a consequence, SQP methods scale very well to large NLP problems with a high number of constraints

Interior Point Methods - overview

- IPM generate iterations that avoid the boundary defined by the inequality constraints
- Each iteration consists of solving a large linear system of equations (the KKT system) taking into account all variables and constraints, so each iteration is fairly expensive
- All three optimality measures of KKT are reduced simultaneously
- As the solver progresses, the iterates are allowed to get closer and closer to the boundary and converge to the optimal solution
- If one tries to solve the KKT system directly, the complementarity condition may turn out to be unsatisfied and may require relaxation, $(\mu g(x^*) = \nu \text{ with } \nu > 0)$, then the relaxation parameter needs to be adjusted in the subsequent iteration
- The IPM perform a relatively small number of expensive iterations; efficient for loosely constrained problems

Unconstrained optimization - Quasi Newton

- An **important element of constraint optimization algorithms** (Quasi-Newton methods: BFGS, DFP, PSB, SR1)

Name		**Condition for updating formula**	**Secant condition**	**Matrix symmetricity**
BFGS	Broyden Fletcher, Goldfarb, Shanno	$min_H \|\|H - H_k\|\|^2$	$H\,y_k = s_k$	$H = H^T$
DFP	Davidon, Fletcher, Powell	$min_B \|\|B - B_k\|\|^2$	$B\,s_k = y_k$	$B = B^T$
PSB	Powell-Symetric-Broyden	$min_B \|\|B - B_k\|\|^2$	$B\,s_k = y_k$	$B - B_k = (B - B_k)^T$
SR1	Symetric-Rank1	$B = B_k + \delta\,\gamma\,\gamma^T$	$B\,s_k = y_k$	
B **– Hessian matrix,** ***H*** **– Hessian inverse,** $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$				

- Nocedal J.,”Theory of Algorithms for Unconstrained Optimization”, Acta Numerica, 1992
- Ding Y., Lushi E., Li Q., ”Investigation of quasi-Newton methods for unconstrained optimization”, Simon Fraser University, Burnaby, B.C. Canada, 2003.

Unconstrained Minimization - BFGS

Proposed independently in 1970 by Broyden, Fletcher, Goldfarb and Shanno - considered **the most efficient quasi-Newton method**

„BFGS method has stood the test of time well and is still regarded as possibly the best secant updating formula"

Nick Gould, 2006, *Rutherford Appleton Laboratory, Chilton, Oxfordshire, England*

Updating formulas in the BFGS method:

The Hessian inverse: $$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

The Hessian matrix: $$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

where $$\rho_k = \frac{1}{y_k^T s_k}$$

Unconstrained minimization for very large problems - conjugate gradients

- **Coniugate gradients** $d_k = -g_k + \beta_k d_{k-1}\,,\quad d_1 = -g_1$
- FR - Fletcher, Reeves………. $\beta_k^{FR} = \|g_k\|^2/\|g_{k-1}\|^2,$
- CD - Coniugate Descent……. $\beta_k^{CD} = -\|g_k\|^2/d_{k-1}^T g_{k-1}\,,$
- DY - Dai-Yuan…………………... $\beta_k^{DY} = \|g_k\|^2/d_{k-1}^T y_{k-1}\,,$
- PRP – Polak-Ribiere-Poliak…… $\beta_k^{PRP} = g_k^T y_{k-1}/\|g_{k-1}\|^2,$
- HS - Hastenes-Stiefel…………… $\beta_k^{HS} = g_k^T y_{k-1}/d_{k-1}^T y_{k-1}\,,$

  Where: $y_{k-1} = g_k - g_{k-1}\,,\qquad \|\cdot\|\ \ \text{2-norm}$

  Reference: Yu-hong Dai, Quin Ni, *Testing different conjugate gradient methods for large scale unconstrained optimization*, J. Comp. Math., 2003

- Comparison based on CUTE problems – PRP & HS seem to be most efficient. Some other 'hybrid' methods are proposed/investigated.

Unconstrained minimization by Sequential Quadratic Approximation (SQA) - Powell’s algorithm NEWUOA

Related Reference:

M.J.D. Powell, „The NEWUOA software for unconstrained optimization without derivatives” *Nonconvex Optimization and Its Applications, Springer US*, **83** (2006).

- **Earlier work on this subject**
  - D. Winfield, „Function Minimization by Interpolation in a Data Table”, *J. Inst. Maths Applics* (1973) **12**, 339-347.
  - A.R. Conn, K. Scheinberg, P.L. Toint, „On the Convergence of Derivative-free methods for Unconstrained Optimization”. *Invited presentation at the Powellfest, Cambridge, July 1996.*

Unconstrained minimization by sequential quadratic approximation - Powell's algorithm NEWUOA

Unconstrained minimization problem min : $F(\underline{x}), \quad \underline{x} \in \mathcal{R}^n$

is replaced by the minimization of $Q_k(\underline{x}) \approx F(\underline{x}), \quad \underline{x} \in \mathcal{R}^n$

Where: $Q(\underline{x}) = c_Q + \underline{g}_Q^T(\underline{x} - \underline{x}_0) + \frac{1}{2}(\underline{x} - \underline{x}_0)^T G_Q (\underline{x} - \underline{x}_0), \quad \underline{x} \in \mathcal{R}^n$

- The model *Q* has *½(n+1)(n+2)* independent parameters,
  This number could be prohibitively expensive when *n* is large
- So, NEWUOA tries to construct quadratic models from fewer data set $\{ \underline{y}^1, \underline{y}^2, \ldots \underline{y}^m \}$, where $(n+2) \leq m \leq \frac{1}{2}(n+1)(n+2)$

$$Q(\underline{y}^i) = F(\underline{y}^i) \quad i = 1, 2, \ldots, m$$

The remaining parameters of *Q* are determined by minimizing the **Frobenius norm** of the difference of two consequtive Hessian models $\|\nabla^2 Q_{\text{new}} - \nabla^2 Q_{\text{old}}\|_F$,

Unconstrained minimization by sequential quadratic approximation - Powell's algorithm NEWUOA (cont.)

- The model *Q* uniquely defined (Frobenius norm strictly convex)
- Lagrange polynomials are used in the interpolation process; parameters of the initial model *Q* calculated easily, owing to convenient selection of the interpolation points:

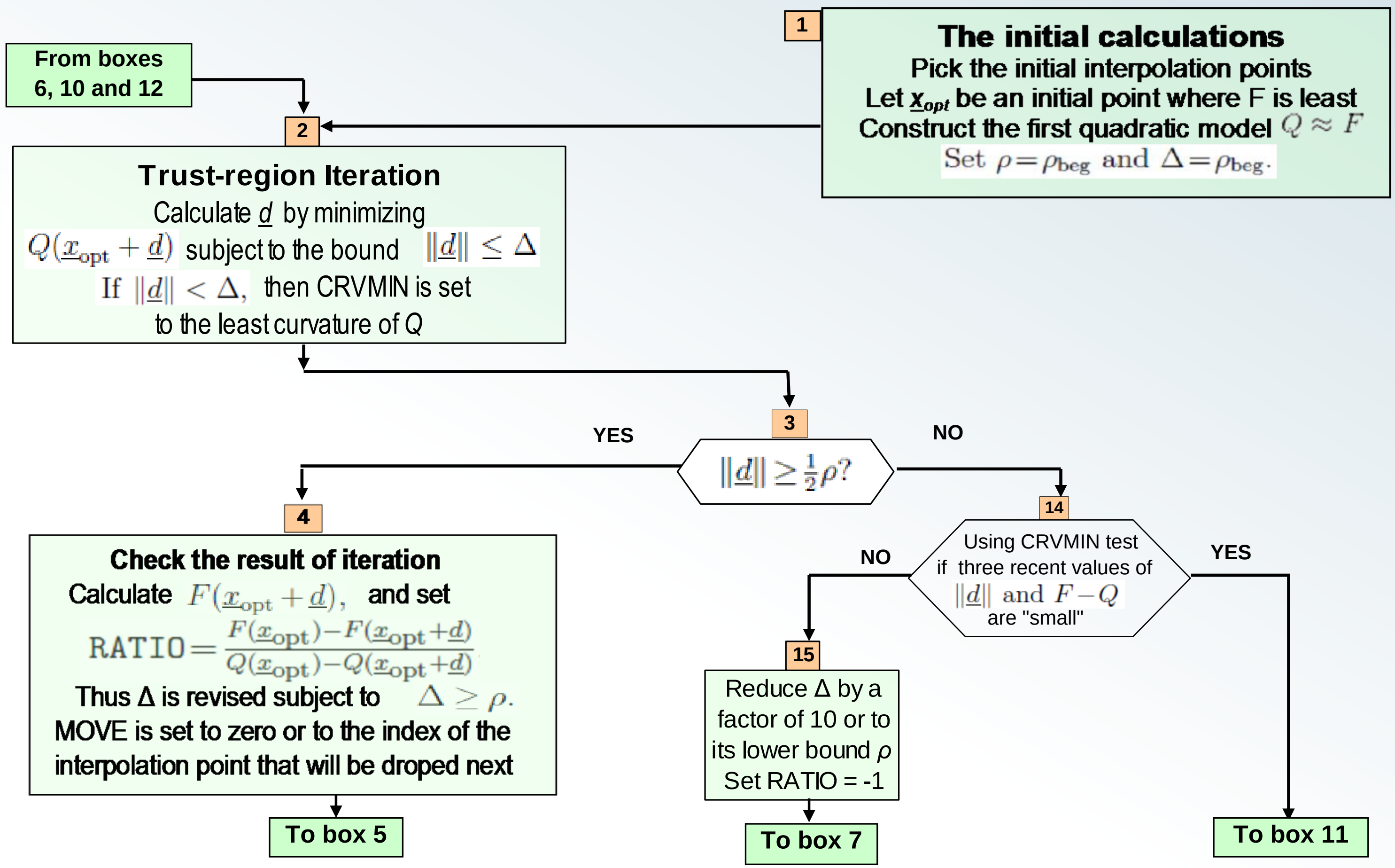
$$\underline{y}^1 = \underline{x}_0, \quad \underline{y}^{i+1} = \underline{x}_0 + \rho_{beg}\,\underline{e}_i\,; \quad \underline{y}^{i+n+1} = \underline{x}_0 - \rho_{beg}\,\underline{e}_i, \quad i = 1, 2, \ldots, n;$$

- The iteration solves the **trust-region quadratic minimization** subproblem using **truncated conjugate gradient method**;

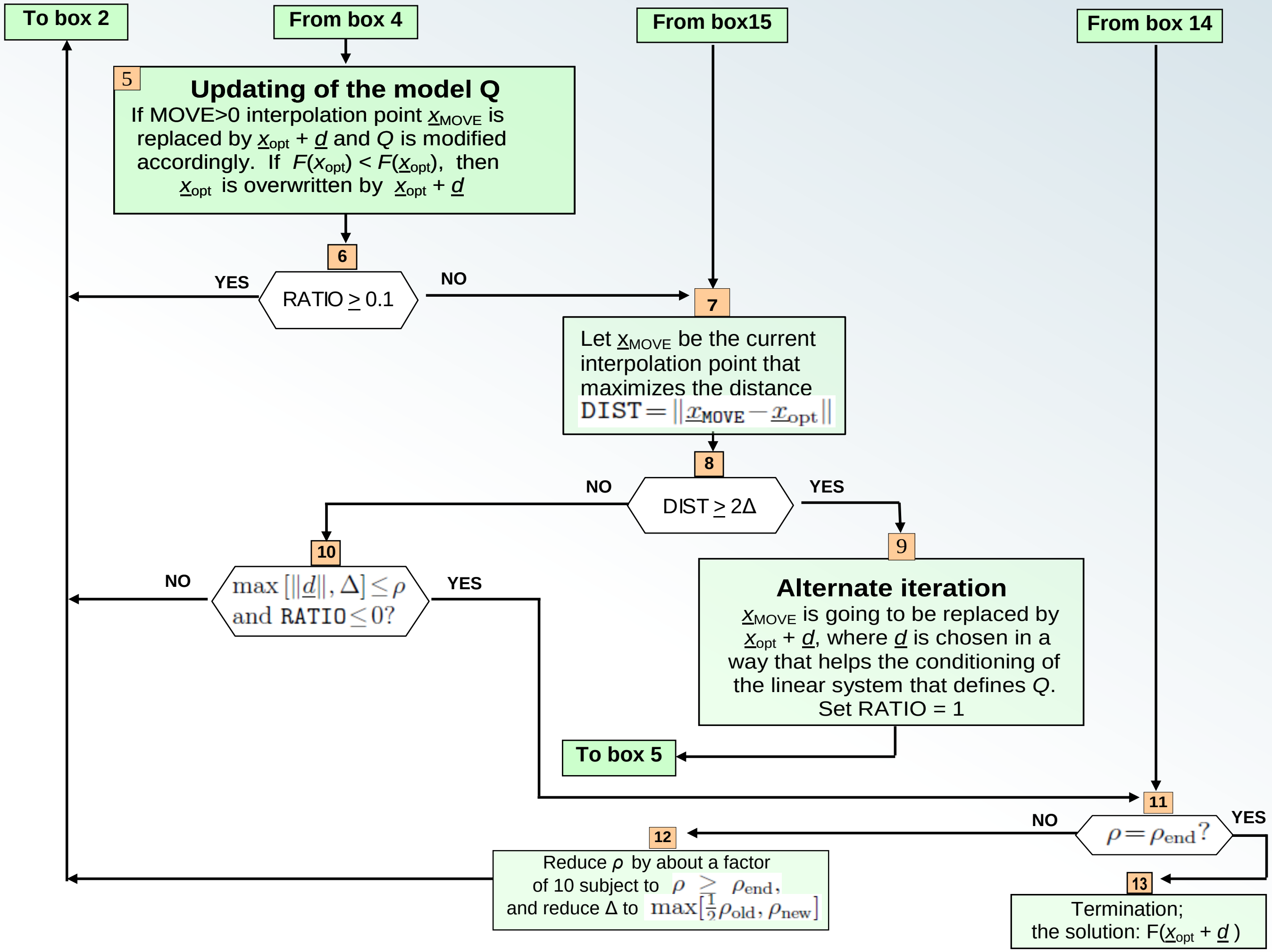
$$\text{Minimize} \quad Q(\underline{x}_{\text{opt}} + \underline{d}) \quad \text{subject to} \quad \|\underline{d}\| \le \Delta,$$

- Only **one point in the interpolation set is replaced** within the iteration; it makes possible to control the **linear independence** of the interpolation condition $Q(\underline{x}^i) = F(\underline{x}^i)$, $i = 1, 2, \ldots, m$;
- *A* special **strategy for selecting the point $\underline{x}^t$ to be replaced** — maximizing the absolute value of the coresponding Lagrangian polynomial (reduce the effect of rounding errors)

Unconstrained minimization by sequential quadratic approximation - Powell's algorithm NEWUOA (cont.)



Unconstrained minimization by sequential quadratic approximation - Powell's algorithm NEWUOA (cont.)



Other algorithms using sequential quadratic approximation

Reference: Powell, M.J.D. (2009). „*The BOBYQA algorithm for bound constrained optimization without derivatives*”, Department of Applied Mathematics and Theoretical Physics, Cambridge University. DAMTP 2009/NA06.

- **BOBYQA -** An iterative algorithm for finding a minimum of a function $F(\underline{x})$, $\underline{x} \in R^n$ subject to bounds $\underline{a} \leq \underline{x} \leq \underline{b}$ on the variables

Reference: Powell, M.J.D. (2014). „*On fast trust region methods for quadratic models with linear constraints*”, Department of Applied Mathematics and Theoretical Physics, Cambridge University. DAMTP 2014/NA02.

- **LINCOA** - An iterative algorithm for finding a minimum of a function $F(\underline{x})$, $\underline{x} \in R^n$ subject to **linear constraints** $\underline{a}_j^T\underline{x} \leq \underline{b}_j$ *using* the trust region framework.
- It is solved by **active set method**, AS may be updated during an iteration, that decreases of freedom in the variables temporarily

An active-set strategy in Powell's algorithm based on sequential quadratic approximation

Reference: Arouxet, B., Echebest, N., Pilotta, E. (2011), „*Active-set strategy in Powells method for optimization without derivatives*", Comp. Appl. Math. Vol. 30.

- An algorithm for solving bound constrained minimization problem without derivatives based on Powell's methods NEWUOA and BOBYQA.
- The algorithm uses trust region framework with **infinity norm** instead of Euclidean norm
- A box constrained problem is solved **using active set strategy** to explore faces of the box. Therefore, it is easily extended to bound constrained minimization problem
- Numerical experiments show that, in general, this alorithm **requires less function evaluations** than Powell's algorithms

How to solve efectively and reliably the constrained NLP problems?

- **Using available ‘off-the-shelf’ solvers ?**

  There are many solvers of SQP and IPM type;

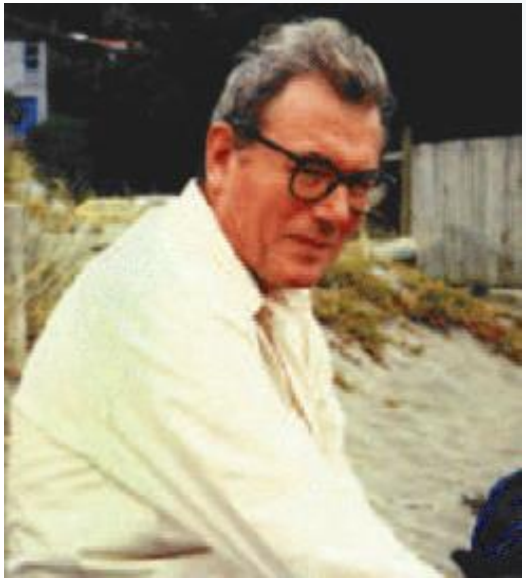
  Suitable solver can be selected according to the specific characteristics of the problem

- **Is a code based on Powell’s penalty function (such as MINCON) still an attractive option ?**

- **How to improve the MINCON code/algorithm** based on the relevant achievements in the field? Is this effort worthwhile ?

Powell's penalty function method with 'shift' as one of the sources of the **augmented Lagrangian method**

What the author thinks about His method?



Since you ask me to mention a gratifying paper, let me pick "A method for nonlinear constraints in minimization problems", because it is regarded as one of the sources of the "augmented Lagrangian method", which is now of fundamental importance in mathematical programming. I have been very fortunate to have played a part in discoveries of this kind.

An Interview with M.J.D. Powell,
Bulletin of the Int. Cent. for Maths, June 2003

Powell’s penalty function method ‘with shift’ (1969) - a kind of Lagrangian function

Penalty function method (Powell, 1969) applied in MINCON

$$\Phi(x,\sigma,\Theta) = F(x) + \sum_{J_1} \sigma_j\,[\varphi_j(x)+\Theta_j]max[0,\ \varphi_j(x)+\Theta_j] + \sum_{J_2}^{m} \sigma_j\,[\psi_j(x)+\Theta_j]^2$$

is similar to the **Augmented Lagrangian method** (see Wierzbicki,1978)

Wierzbicki,A., ‘A_quadratic_approximation_method_based on_augmented Lagrangian functions for nonconvex nonlineat programming problems’, IIASA, Laxenburg, Austria (1978)

„the problem of finding an adequate **penalty shift** (θ_j^*) is equivalent to the fundamental problem of finding Lagrangian multiplier...” ($\underline{\lambda}_j^* = \sigma_j\ \theta_j^*$)

*„..In practical applications, the algorithm is **very robust**, it is rather difficult to find practical problems for which this algorithm does not work, as long as the required accuracy is not too high...”*

(the presence of ‘weakly active constraints’ at the solution $\underline{x}^*$ results in discontinuities of $\Phi(\underline{x}^*,\ \sigma_j^*,\ \theta_j^*$; it may generate numerical problems).

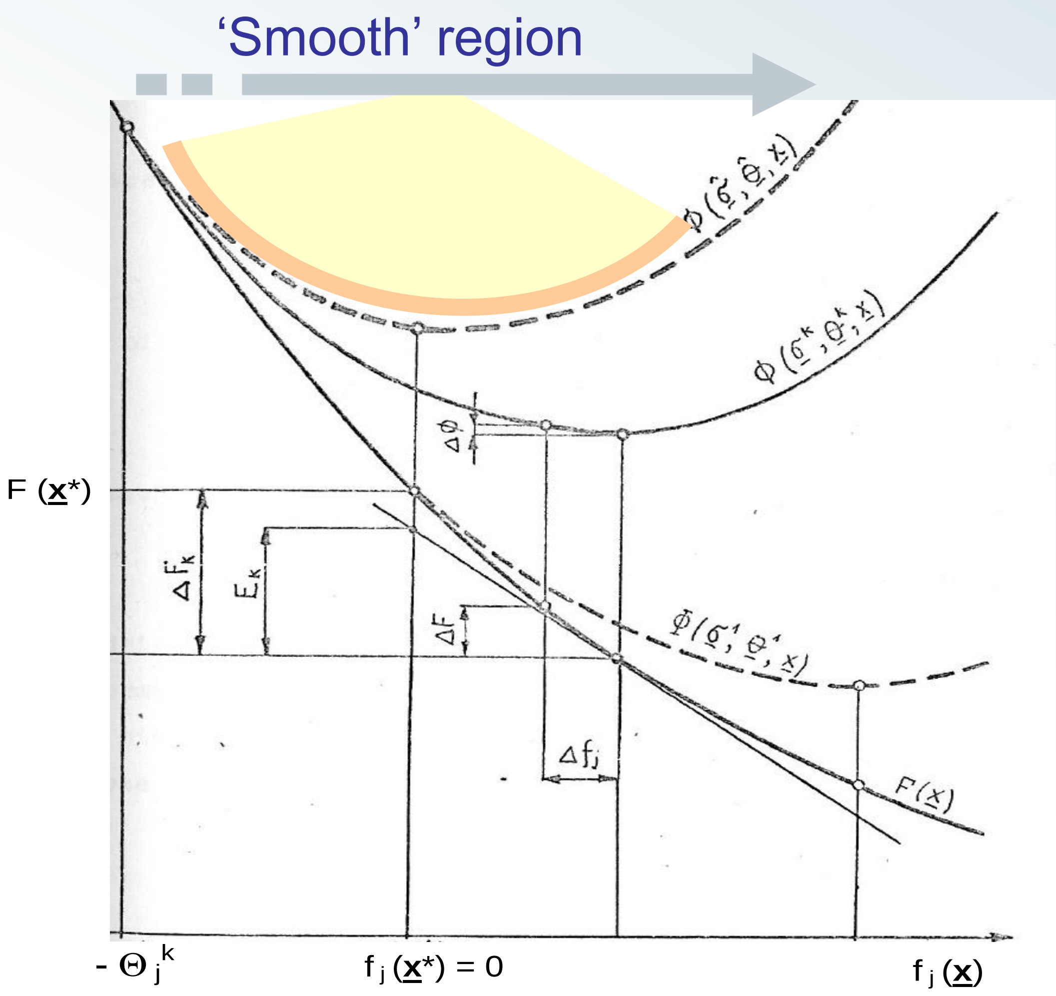
How the MINCON code/algorithm could be improved?

- **Unconstrained minimization** solver based on Davidon's method can be replaced by a more robust/efficient solver
  - **Consideration can be given to BFGS** (considered one of the best solvers)
    - This option still requires **numerical approximation of the gradient** with all numerical problems (rounding off/ truncation errors),
    - Potential numerical problems (illconditioning) in the early stage of the iterative process when the function may be locally non-convex)
    - Positive feature of using VM method - approximating the gradient may be made less laborious, when the algorithm used for calculating the model has a block structure.

How the MINCON code/algorithm could be improved?

- **Use of the SQA unconstrained optimization solver** NEWBYQA (Powell, 2006) for minimizing the penalty function

  - Quadratic approximation of the model and the use of trust region framework, with other safeguards incorporated in the NEWBYQA algorithm are expected to reduce **negative effects of rounding off and truncation** errors;
  - A good **approximation of Hessian matrix (psd)** at the solution x* generated by NEWBYQA can be used in the subsequent constrained (large) iterations
  - In the final phase the iterations are likely to avoid a **non-smooth region** of the penalty function *Φ(x)*, (the parameters $\theta_j = y_j^*/\sigma_j$ and σ_j can be kept moderate; see Wierzbicki, 1978); therefore, potential effects of discontinuities in the Hessian should not be a problem.



$$\theta_i = y_i^* / \sigma_i$$

Useful Bibliography

- Wierzbicki, A., „A Quadratic Approximation Method Based on Augmented Lagrangian Functions for Nonconvex Nonlinear Programming Problems, IIASA, Laxenburg, Austria (1978)
- Rockafellar, R T, Lagrange Multipliers And Optimality, 1993. https://web.williams.edu/Mathematics/sjmiller/public_html/105Sp10/handouts/Rockafellar_LagrangeMultAndOptimality.pdf
- Klain, D, Lagrange-multipliers without permanent scarring. http://nlp.cs.berkeley.edu/tutorials/lagrange-multipliers.pdf
- Shewchuk J R, Conjugate gradient without the agonizing pain,1994
- Conn A.R., Gould N., Toint P., "Trust-Region Methods" (MPS-SIAM Series on Optimization), 2000; https://epdf.tips/queue/trust-region-methods-mps-siam-series-on-optimizatione1bc3616eccf9dda948f1d0423f2a2b875221.html
- Powell, M.J.D. (2009). „The BOBYQA algorithm for bound constrained optimization without derivatives", Department of Applied Mathematics and Theoretical Physics, Cambridge University. DAMTP 2009/NA06.
- Ding Y, Lushi E, Li Q, Investigation of quasi-Newton methods for unconstrained optimization, Simon Fraser University, Burnaby, B.C. Canada, 2003. http://people.math.sfu.ca/~elushi/project_833.pdf

enco

Useful Bibliography

- Nocedal J.,"Theory of Algorithms for Unconstrained Optimization" Acta Numerica 1992 http://users.iems.northwestern.edu/~nocedal/PDFfiles/acta.pdf
- Vanderbei, S., 2016, Interior Point Method for Nonconvex Programming_predcor https://vanderbei.princeton.edu/ps/predcor.pdf
- MJD Powell, „On fast trust region methods for quadratic models with linear constraints", DAMTP 2014 NA02. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2014_02.pdf
- Conn A R, Scheinberg K, Toint P. L., „On the convergence of derivative-free methods for unconstrained optimization", 1996. https://www.researchgate.net/profile/Andrew_Conn/publication/240151483_On_the_Convergence_of_Derivative-Free_Methods_for_Unconstrained_Optimization/links/02e7e51c85a93acb52000000.pdf
- Yuan Y., „On the truncated conjugate gradient method", 1999. ftp://lsec.cc.ac.cn/pub/home/yyx/papers/p989-net.pdf
- K. Schittkowski, Ya-xiang Yuan, "Sequential Quadratic Programming Methods", Wiley Encyclopedia of Operations Research and Management Science (2011), preprint; http://www.klaus-schittkowski.de/SQP_review.htm
- Sachsenberg B. , Schittkowski K., „A Combined SQP-IPM Algorithm for Solving Large-Scale Nonlinear Optimization Problems", Optimization Letters Vol. 9, 1271-1282 (2015); http://www.klaus-schittkowski.de/nlpip_report.htm

enco