The Mathematical legacy of ECOSSE

Julian Hall

School of Mathematics

University of Edinburgh

With help from Sven Leyffer

April 17th 2009

Mathematics within ECOSSE

- Centred on Roger Fletcher's group in Dundee

Mathematics within ECOSSE

- Centred on Roger Fletcher's group in Dundee
- Other work done in Edinburgh under Ken McKinnon

Mathematics within ECOSSE

- Centred on Roger Fletcher's group in Dundee
- Other work done in Edinburgh under Ken McKinnon
- Engineers were also using Maths!

Who am I?

Who am I?

- First on Jack's list!

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!
- Worked on a Chemical Engineering problem

  Flexible retrofit design of multiproduct batch plants
  Fletcher, Hall and Johns (1991)
  Computers and Chemical Engineering **15** 843–852

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!
- Worked on a Chemical Engineering problem

  Flexible retrofit design of multiproduct batch plants
  Fletcher, Hall and Johns (1991)
  Computers and Chemical Engineering **15** 843–852

- Otherwise, we took the money and ran...

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!
- Worked on a Chemical Engineering problem

  Flexible retrofit design of multiproduct batch plants
  Fletcher, Hall and Johns (1991)
  Computers and Chemical Engineering **15** 843–852

- Otherwise, we took the money and ran...
- and went hill-walking...

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!
- Worked on a Chemical Engineering problem

  Flexible retrofit design of multiproduct batch plants
  Fletcher, Hall and Johns (1991)
  Computers and Chemical Engineering **15** 843–852

- Otherwise, we took the money and ran...
- and went hill-walking...
- and did maths...

Who am I?

- First on Jack's list!
- Doctoral research assistant to Roger Fletcher (1988–1990)
- Saved from the penury of an SERC studentship. Thanks Jack!
- Worked on a Chemical Engineering problem

  Flexible retrofit design of multiproduct batch plants
  Fletcher, Hall and Johns (1991)
  Computers and Chemical Engineering **15** 843–852

- Otherwise, we took the money and ran...
- and went hill-walking...
- and did maths...
- and paved the way for the German invasion

The Dundee group

- Major focus was the solution of mixed-integer nonlinear programming **MINLP** problems

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to} & \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{0} \\ & y_i \in \{0,\ 1\}\ \forall i \end{array}$$

- **Why?**

The Dundee group

- Major focus was the solution of mixed-integer nonlinear programming **MINLP** problems

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}, \boldsymbol{y}) \\ &\text{subject to } \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{0} \\ &\qquad\qquad y_i \in \{0,\ 1\}\ \forall i \end{aligned}$$

- **Why?**
  Retrofit design problems required the solution of (convex) MINLP problems

The Dundee group

- Major focus was the solution of mixed-integer nonlinear programming **MINLP** problems

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}, \boldsymbol{y}) \\ &\text{subject to } \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{0} \\ &\qquad\qquad y_i \in \{0,\ 1\}\ \forall i \end{aligned}$$

- **Why?**
  Retrofit design problems required the solution of (convex) MINLP problems
- What is required in order to solve MINLP problems?

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

↓

Solve **NLP**$^{(k)}$

minimize $f(\boldsymbol{x},\ \boldsymbol{y}^{(k)})$

s.t. $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y}^{(k)}) \geq \boldsymbol{0}$

to obtain $\boldsymbol{x}^{(k)}$

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

Solve **NLP**$^{(k)}$

minimize $f(\boldsymbol{x},\ \boldsymbol{y}^{(k)})$

s.t. $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y}^{(k)}) \geq \boldsymbol{0}$

to obtain $\boldsymbol{x}^{(k)}$

Linearise $f(\boldsymbol{x},\ \boldsymbol{y})$ and $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y})$ about $(\boldsymbol{x}^{(k)},\ \boldsymbol{y}^{(k)})$

to form $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$

from $A^{(k-1)}$, $B^{(k-1)}$, $\boldsymbol{g}^{(k-1)}$ and $\boldsymbol{b}^{(k-1)}$

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

Solve **NLP**$^{(k)}$
minimize $f(\boldsymbol{x},\ \boldsymbol{y}^{(k)})$
s.t. $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y}^{(k)}) \geq \boldsymbol{0}$
to obtain $\boldsymbol{x}^{(k)}$

Linearise $f(\boldsymbol{x},\ \boldsymbol{y})$ and $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y})$ about $(\boldsymbol{x}^{(k)},\ \boldsymbol{y}^{(k)})$
to form $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$
from $A^{(k-1)}$, $B^{(k-1)}$, $\boldsymbol{g}^{(k-1)}$ and $\boldsymbol{b}^{(k-1)}$

Solve **MILP**$^{(k)}$
minimize z
s.t. $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$
$y_i \in \{0,\ 1\}\ \forall i$
to obtain $\boldsymbol{y}^{(k+1)}$

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

Solve **NLP**$^{(k)}$
minimize $f(\boldsymbol{x},\ \boldsymbol{y}^{(k)})$
s.t. $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y}^{(k)}) \geq \boldsymbol{0}$
to obtain $\boldsymbol{x}^{(k)}$

Linearise $f(\boldsymbol{x},\ \boldsymbol{y})$ and $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y})$ about $(\boldsymbol{x}^{(k)},\ \boldsymbol{y}^{(k)})$
to form $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$
from $A^{(k-1)}$, $B^{(k-1)}$, $\boldsymbol{g}^{(k-1)}$ and $\boldsymbol{b}^{(k-1)}$

Solve **MILP**$^{(k)}$
minimize z
s.t. $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$
$y_i \in \{0,\ 1\}\ \forall i$
to obtain $\boldsymbol{y}^{(k+1)}$

$k := k + 1$

Outer-approximation: Duran and Grossmann (1986)

Choose $\boldsymbol{y}^{(0)}$; $k = 0$

Solve **NLP**$^{(k)}$

minimize $f(\boldsymbol{x},\ \boldsymbol{y}^{(k)})$

s.t. $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y}^{(k)}) \geq \boldsymbol{0}$

to obtain $\boldsymbol{x}^{(k)}$

Linearise $f(\boldsymbol{x},\ \boldsymbol{y})$ and $\boldsymbol{c}(\boldsymbol{x},\ \boldsymbol{y})$ about $(\boldsymbol{x}^{(k)},\ \boldsymbol{y}^{(k)})$

to form $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$

from $A^{(k-1)}$, $B^{(k-1)}$, $\boldsymbol{g}^{(k-1)}$ and $\boldsymbol{b}^{(k-1)}$

Solve **MILP**$^{(k)}$

minimize z

s.t. $A^{(k)}\boldsymbol{x} + B^{(k)}\boldsymbol{y} + \boldsymbol{g}^{(k)}z \geq \boldsymbol{b}^{(k)}$

$y_i \in \{0,\ 1\}\ \forall i$

to obtain $\boldsymbol{y}^{(k+1)}$

$k := k + 1$

Components of the first Dundee MINLP solver

- NLP solver from NAG

Components of the first Dundee MINLP solver

- NLP solver from NAG
- MILP solver

Components of the first Dundee MINLP solver

- NLP solver from NAG
- MILP solver
  - Branch-and-bound by Hall

Components of the first Dundee MINLP solver

- NLP solver from NAG
- MILP solver
  - Branch-and-bound by Hall
  - "Reliable" simplex LP solver by Fletcher

Components of the first Dundee MINLP solver

- NLP solver from NAG
- MILP solver
  - Branch-and-bound by Hall
  - "Reliable" simplex LP solver by Fletcher
  - Sparse matrix algebra by Hall

Components of the first Dundee MINLP solver

- NLP solver from NAG
- MILP solver
  - Branch-and-bound by Hall
  - "Reliable" simplex LP solver by Fletcher
  - Sparse matrix algebra by Hall
- *Towards reliable linear programming*
  Fletcher and Hall (1990)
  Pitman Research Notes in Mathematics Series 228 89–104
- *Ordering algorithms for irreducible sparse linear systems*
  Fletcher and Hall (1993)
  Annals of Operations Research **43** 15–32

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

  Won the 2005 COAP best paper prize

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)
- Still working on (parallelising) the revised simplex method

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)
- Still working on (parallelising) the revised simplex method
- Consultancy work in the following software industries

  Animal feed,

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)
- Still working on (parallelising) the revised simplex method
- Consultancy work in the following software industries

  Animal feed, Petroleum,

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)
- Still working on (parallelising) the revised simplex method
- Consultancy work in the following software industries

  Animal feed, Petroleum, Power and water,

After ECOSSE?

- Came to Edinburgh as a Maths lecturer (1990–date)
- With Ken McKinnon
  - Wrote a revised simplex solver using improved sparse matrix algebra (1992–2001)
  - Identifying and exploiting hyper-sparsity was a major revised simplex development

    Hyper-sparsity in the revised simplex method and how to exploit it
    Hall and McKinnon (2005)
    Computational Optimization and Applications **32(3)** 259–283

    Won the 2005 COAP best paper prize
  - Developed and implemented parallel revised simplex schemes (1994–97)
- With Jacek Gondzio (Edinburgh)
  - Applied simplex algebra to interior-point methods (2005–06)
- Still working on (parallelising) the revised simplex method
- Consultancy work in the following software industries

  Animal feed, Petroleum, Power and water, Chemical engineering

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

What is hyper-sparsity

Consider solving $\boldsymbol{Ax} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

Consequence:

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

Consequence:

Given sparse $LU = A$, solving $A\boldsymbol{x} = \boldsymbol{b}$ by traditional substitution can be very inefficient

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

Consequence:

Given sparse $LU = A$, solving $A\boldsymbol{x} = \boldsymbol{b}$ by traditional substitution can be very inefficient

When does hyper-sparsity occur?

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

Consequence:

Given sparse $LU = A$, solving $A\boldsymbol{x} = \boldsymbol{b}$ by traditional substitution can be very inefficient

When does hyper-sparsity occur?

- During left-looking implementations of Gaussian elimination

What is hyper-sparsity

Consider solving $A\boldsymbol{x} = \boldsymbol{b}$ when A is **sparse**

- If $\boldsymbol{b}$ is **dense** then $\boldsymbol{x}$ is **dense**
- If $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is usually **dense**—A^{-1} is usually dense
- If A^{-1} is **sparse** then A is **hyper-sparse**
- If A is **hyper-sparse** and $\boldsymbol{b}$ is **sparse** then $\boldsymbol{x}$ is **sparse**

Consequence:

Given sparse $LU = A$, solving $A\boldsymbol{x} = \boldsymbol{b}$ by traditional substitution can be very inefficient

When does hyper-sparsity occur?

- During left-looking implementations of Gaussian elimination
- When applying the revised simplex method to important classes of LP problems

Exploiting hyper-sparsity

- Traditional technique for solving $L\boldsymbol{x} = \boldsymbol{b}$ by transforming $\boldsymbol{b}$ into $\boldsymbol{x}$

```
do k = 1, n
    b := b − b_k l_k
end do
```

do $k = 1$, n

$\quad \boldsymbol{b} := \boldsymbol{b} - b_k \boldsymbol{l}_k$

end do

Exploiting hyper-sparsity

- Traditional technique for solving $L\boldsymbol{x} = \boldsymbol{b}$ by transforming $\boldsymbol{b}$ into $\boldsymbol{x}$

```
do k = 1, n
   b := b − b_k l_k
end do
```

- When $\boldsymbol{b}$ is sparse skip $\boldsymbol{l}_k$ if b_k is zero

```
do k = 1, n
   if (b_k .ne. 0) then
      b := b − b_k l_k
   end if
end do
```

Exploiting hyper-sparsity

- Traditional technique for solving $\boldsymbol{Lx} = \boldsymbol{b}$ by transforming $\boldsymbol{b}$ into $\boldsymbol{x}$

```
do k = 1, n
   b := b − b_k l_k
end do
```

- When $\boldsymbol{b}$ is sparse skip $\boldsymbol{l}_k$ if b_k is zero

```
do k = 1, n
   if (b_k .ne. 0) then
      b := b − b_k l_k
   end if
end do
```

- When $\boldsymbol{x}$ is sparse, the dominant cost is the test for zero

Exploiting hyper-sparsity

- Traditional technique for solving $\boldsymbol{Lx} = \boldsymbol{b}$ by transforming $\boldsymbol{b}$ into $\boldsymbol{x}$

do $k = 1,\ n$
    $\boldsymbol{b} := \boldsymbol{b} - b_k\boldsymbol{l}_k$
end do

- When $\boldsymbol{b}$ is sparse skip $\boldsymbol{l}_k$ if b_k is zero

do $k = 1,\ n$
    if (b_k .ne. 0) then
        $\boldsymbol{b} := \boldsymbol{b} - b_k\boldsymbol{l}_k$
    end if
end do

- When $\boldsymbol{x}$ is sparse, the dominant cost is the test for zero
- Efficient identification of vectors $\boldsymbol{l}_k$ to be applied
  - Gilbert and Peierls (1988)
  - Hall and McKinnon (1998–2005)

The Leyffer years (1989–2002)

The Leyffer years (1989–2002)

- MIQP

  Numerical Experience with lower bounds for MIQP branch–and–bound
  Fletcher and Leyffer (1998)
  SIAM J. Optimization **8(2)** 604–616

The Leyffer years (1989–2002)

- MIQP

  Numerical Experience with lower bounds for MIQP branch–and–bound
  Fletcher and Leyffer (1998)
  SIAM J. Optimization **8(2)** 604–616

- MINLP

The Leyffer years (1989–2002)

- MIQP

  Numerical Experience with lower bounds for MIQP branch–and–bound
  Fletcher and Leyffer (1998)
  SIAM J. Optimization **8(2)** 604–616

- MINLP
  - Replaced MILP master problems by MIQP

The Leyffer years (1989–2002)

- MIQP

  Numerical Experience with lower bounds for MIQP branch–and–bound
  Fletcher and Leyffer (1998)
  SIAM J. Optimization **8(2)** 604–616

- MINLP
  - Replaced MILP master problems by MIQP
  - Generalised outer approximation

    Solving mixed integer nonlinear programs by outer approximation
    Fletcher and Leyffer (1994)
    Mathematical Programming **66** 327–349

The Leyffer years (1989–2002)

- MIQP

  Numerical Experience with lower bounds for MIQP branch–and–bound
  Fletcher and Leyffer (1998)
  SIAM J. Optimization **8(2)** 604–616

- MINLP
  - Replaced MILP master problems by MIQP
  - Generalised outer approximation

    Solving mixed integer nonlinear programs by outer approximation
    Fletcher and Leyffer (1994)
    Mathematical Programming **66** 327–349

  - Integrated NLP within branch-and-bound

    Integrating SQP and branch-and-bound for mixed integer nonlinear programming
    Leyffer (2001)
    Computational Optimization & Applications **18** 295–309

The Leyffer years (1989–2002)

- Eventually did some chemical engineering!

  Comparison of certain MINLP algorithms when applied to a model structure determination and parameter estimation problem
  Skrifvars, Leyffer and Westerlund (1998)
  Computers and Chemical Engineering **22(12)** 1829–1835

The “Filter” method

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$$

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \mathbf{0}$$

- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^{-}\|$

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \mathbf{0}$$

- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^-\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \mathbf{0}$$

- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^{-}\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \mathbf{0}$$

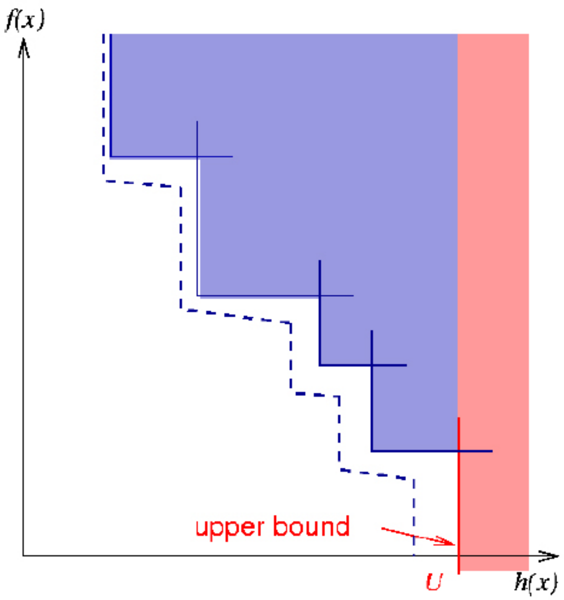
- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^{-}\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$$

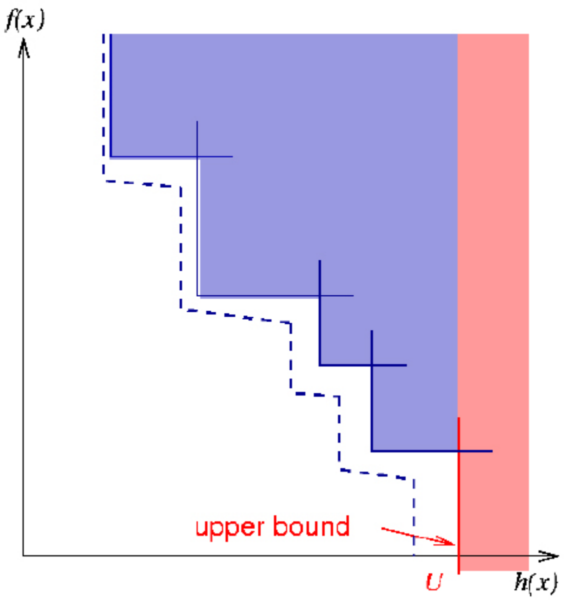
- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^-\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach



The "Filter" method

For NLP

minimize $f(\boldsymbol{x})$ subject to $\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$



- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^-\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach

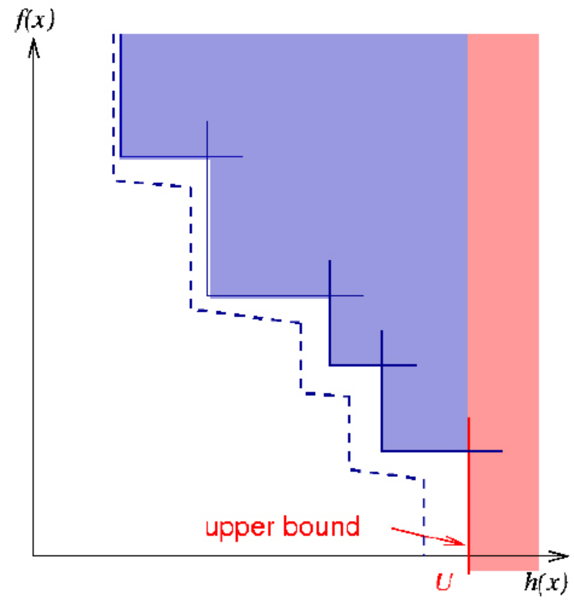
Where did the idea come from?

The "Filter" method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$$

- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^-\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach



Where did the idea come from?

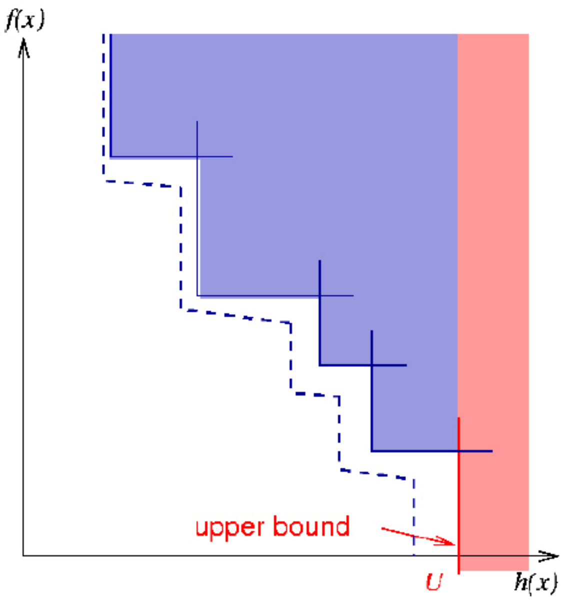
- Used by Radcliffe and Surry (Edinburgh) with genetic algorithms (1995)

The “Filter” method

For NLP

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$$

- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^{-}\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach



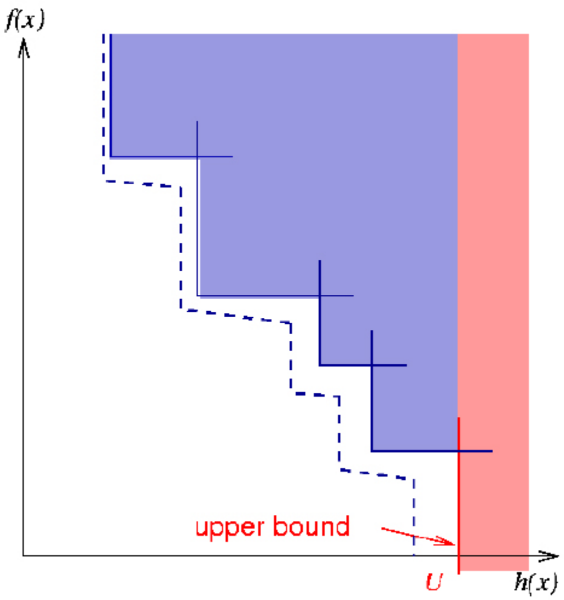
Where did the idea come from?

- Used by Radcliffe and Surry (Edinburgh) with genetic algorithms (1995)
- First presented by Leyffer at an ECOSSE meeting (1996)

The "Filter" method

For NLP

minimize $f(\boldsymbol{x})$ subject to $\boldsymbol{c}(\boldsymbol{x}) \geq \mathbf{0}$



- Want to minimize *both* $f(\boldsymbol{x})$ *and* $h(\boldsymbol{x}) = \|\boldsymbol{c}(\boldsymbol{x})^-\|$
- **Traditionally:** minimize single objective $f(\boldsymbol{x}) + \pi h(\boldsymbol{x})$
- Hard to identify penalty parameter π
- Filter avoided this using a multi-objective approach

Where did the idea come from?

- Used by Radcliffe and Surry (Edinburgh) with genetic algorithms (1995)
- First presented by Leyffer at an ECOSSE meeting (1996)
- First presented by Fletcher in a plenary talk at SIAM Optimization (May 1996)

Legacy

- Convergence proof and extensions generated a sequence of landmark papers
  - *Nonlinear programming without a penalty function*
    Fletcher and Leyffer (2002)
    Mathematical Programming **91** 239–270
  - *On the global convergence of a filter-SQP algorithm*
    Fletcher, Leyffer and Toint (2002)
    SIAM J. Optimization **13(1)** 44–59
  - *Global convergence of trust-region SQP-filter algorithms for general nonlinear programming*
    Fletcher, Gould, Leyffer, Toint, and Wächter (2002)
    SIAM J. Optimization **13(3)** 635–659

Legacy

- Convergence proof and extensions generated a sequence of landmark papers
  - *Nonlinear programming without a penalty function*
    Fletcher and Leyffer (2002)
    Mathematical Programming **91** 239–270
  - *On the global convergence of a filter-SQP algorithm*
    Fletcher, Leyffer and Toint (2002)
    SIAM J. Optimization **13(1)** 44–59
  - *Global convergence of trust-region SQP-filter algorithms for general nonlinear programming*
    Fletcher, Gould, Leyffer, Toint, and Wächter (2002)
    SIAM J. Optimization **13(3)** 635–659
- Extended (with Fletcher, Gould and Toint) to algebraic inequalities, nonlinear systems of equations and nonlinear least squares

Legacy

- Convergence proof and extensions generated a sequence of landmark papers
  - *Nonlinear programming without a penalty function*
    Fletcher and Leyffer (2002)
    Mathematical Programming **91** 239–270
  - *On the global convergence of a filter-SQP algorithm*
    Fletcher, Leyffer and Toint (2002)
    SIAM J. Optimization **13(1)** 44–59
  - *Global convergence of trust-region SQP-filter algorithms for general nonlinear programming*
    Fletcher, Gould, Leyffer, Toint, and Wächter (2002)
    SIAM J. Optimization **13(3)** 635–659
- Extended (with Fletcher, Gould and Toint) to algebraic inequalities, nonlinear systems of equations and nonlinear least squares
- With Fletcher and Toint, won the SIAM Lagrange prize in 2006

After ECOSSE?

- Work on mathematical programs with equilibrium constraints (MPEC) was motivation by Bill Morton's multi-phase flow example presented at an ECOSSE meeting)

After ECOSSE?

- Work on mathematical programs with equilibrium constraints (MPEC) was motivation by Bill Morton's multi-phase flow example presented at an ECOSSE meeting)
- Went to Argonne as a researcher (2002–date)

After ECOSSE?

- Work on mathematical programs with equilibrium constraints (MPEC) was motivation by Bill Morton's multi-phase flow example presented at an ECOSSE meeting)
- Went to Argonne as a researcher (2002–date)
- Applies mathematical programming to a range of problems

After ECOSSE?

- Work on mathematical programs with equilibrium constraints (MPEC) was motivation by Bill Morton's multi-phase flow example presented at an ECOSSE meeting)
- Went to Argonne as a researcher (2002–date)
- Applies mathematical programming to a range of problems
- Makes up wacky titles for conference presentations

The Return of the Filter	Dundee/SIAM
How the Grinch Solved MPECs	Dundee
Filter Methods By Dummies (*)	Argonne
MPECs: Much Ado About Nothing?	ISMP-00
The M&M of Optimization: Modeling and Methods	Argonne
QPPAL: A Friendly Solver for Large-Scale QPs	INFORMS

(*) actually a misprint: should have been "For"

The rest of the German invasion

- Christine Zoppke (1991–95)

The rest of the German invasion

- Christine Zoppke (1991–95)
  - Worked on “tolerance tubes” (unpublished)

The rest of the German invasion

- Christine Zoppke (1991–95)
  - Worked on “tolerance tubes” (unpublished)
  - Led to filter “to some extent”

The rest of the German invasion

- Christine Zoppke (1991–95)
  - Worked on “tolerance tubes” (unpublished)
  - Led to filter “to some extent”
  - Rediscovered as a “funnel” by Gould and Toint

The rest of the German invasion

- Christine Zoppke (1991–95)
  - Worked on “tolerance tubes” (unpublished)
  - Led to filter “to some extent”
  - Rediscovered as a “funnel” by Gould and Toint
- Not funded by ECOSSE, but part of the “scene”
  - Frank Plab (Dundee 1989–90; Edinburgh 1990-99)
    Early work on parallel revised simplex

The rest of the German invasion

- Christine Zoppke (1991–95)
  - Worked on "tolerance tubes" (unpublished)
  - Led to filter "to some extent"
  - Rediscovered as a "funnel" by Gould and Toint
- Not funded by ECOSSE, but part of the "scene"
  - Frank Plab (Dundee 1989–90; Edinburgh 1990-99)
    Early work on parallel revised simplex
  - Andreas Grothey (Dundee 1994–96; Edinburgh 1996–date)
    Serial and parallel interior point

Meanwhile, in Edinburgh...

Meanwhile, in Edinburgh...

- Under Ken McKinnon
  - Marcel Mongeau: post-doctoral research assistant (1993–94)
  - Colin Millar: masters research assistant (1994–1998)

Meanwhile, in Edinburgh...

- Under Ken McKinnon
  - Marcel Mongeau: post-doctoral research assistant (1993–94)
  - Colin Millar: masters research assistant (1994–1998)
- Worked on applications of optimization in chemical engineering

Meanwhile, in Edinburgh...

- Under Ken McKinnon
  - Marcel Mongeau: post-doctoral research assistant (1993–94)
  - Colin Millar: masters research assistant (1994–1998)
- Worked on applications of optimization in chemical engineering
  - *A generic global optimization algorithm for the chemical and phase equilibrium problem*
    McKinnon and Mongeau (1998)
    Journal of Global Optimization **12(4)** 325–351
  - *Global optimization for the chemical and phase equilibrium problem using interval analysis*
    McKinnon, Millar and Mongeau (1996)
    In State of the Art in Global Optimization: Computational Methods and Applications 365–382

Meanwhile, in Edinburgh...

- Under Ken McKinnon
  - Marcel Mongeau: post-doctoral research assistant (1993–94)
  - Colin Millar: masters research assistant (1994–1998)
- Worked on applications of optimization in chemical engineering
  - *A generic global optimization algorithm for the chemical and phase equilibrium problem*
    McKinnon and Mongeau (1998)
    Journal of Global Optimization **12(4)** 325–351
  - *Global optimization for the chemical and phase equilibrium problem using interval analysis*
    McKinnon, Millar and Mongeau (1996)
    In State of the Art in Global Optimization: Computational Methods and Applications 365–382
- Where are they now?
  - Marcel Mongeau: maître de conférences at Université Paul Sabatier, Toulouse
  - Colin Millar: DJ?

Future work?

Future work?

Differential equation constrained optimization

Future work?

Differential equation constrained optimization

- Part of an early (pre-1990) planning document

Future work?

Differential equation constrained optimization

- Part of an early (pre-1990) planning document
- Now a major field of optimization

Conclusions

- Funding mathematicians isn't a good way to do Chemical Engineering research

Conclusions

- Funding mathematicians isn't a good way to do Chemical Engineering research
- Asking them to solve MINLP problems was a good idea!

Conclusions

- Funding mathematicians isn't a good way to do Chemical Engineering research
- Asking them to solve MINLP problems was a good idea!
- ECOSSE was the springboard for some outstanding optimization research

Conclusions

- Funding mathematicians isn't a good way to do Chemical Engineering research
- Asking them to solve MINLP problems was a good idea!
- ECOSSE was the springboard for some outstanding optimization research

Thanks Jack!