Hybrid methods for optimisation

INYS Workshop, Vilnius

Eric S Fraga

e.fraga@ucl.ac.uk

Department of Chemical Engineering
University College London
Motivation

Need to provide robust optimization for process design with confidence in results:

- complex non-linear, non-convex, discontinuous & noisy models.
- small, possibly non-convex, feasible regions
- ill- or un-defined objective function and constraint equations outside feasible regions

One possible solution is the use of a combinations of methods to achieve good results under these adverse conditions.
Hybrid methods

Hybrid (*Hy"brid*), a.
derived by a mixture of characteristics from two distinctly different sources;

The GNU version of The Collaborative International Dictionary of English

Two approaches for combining methods:

- embedded
- sequential (or *multi-start*)
## Embedded approaches

<table>
<thead>
<tr>
<th>Outer Method</th>
<th>Inner Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backtracking</td>
<td>Local search</td>
</tr>
<tr>
<td>Genetic Algorithm (GA)</td>
<td>Simulated Annealing (SA)</td>
</tr>
<tr>
<td>Random walk</td>
<td>Conflict resolution</td>
</tr>
<tr>
<td>Tabu</td>
<td>Iterated local search</td>
</tr>
<tr>
<td>Ant colony (ACO)</td>
<td>Constraint propagation</td>
</tr>
<tr>
<td>Direct search</td>
<td>Genetic Algorithm</td>
</tr>
</tbody>
</table>

Successful for a large range of problems but typically requires targeted procedures and/or models.
Sequential hybrid methods

Basic procedure:

- A method is applied to obtain a solution.
- This solution is used as an initial point for another method.
- Procedure can iterate or be extended with further methods.
- Essentially any combination of methods could be considered.

The remainder of this talk presents two case studies which illustrate the usefulness of sequential hybrid methods.
Case study: Water distribution networks

Given

- Layout (connectivity, length, $L_k$, alternative discrete pipe diameters)
- Node demands, $D_n$
- Minimum head requirements, $H_n^{\text{min}}$

Determine

- diameter of each pipe, $d_k$
- flow amount and direction, $Q_k$
- head (pressure) at each node, $H_n$

so as to minimise total network cost.

**Notation:** $k$ index for pipes/connections and $n$ index for nodes.
7 node, 8 pipe, 1 reservoir example without pumps 
The model

\[ \min \sum_{k} \sum_{m} C_{m}L_{k}y_{km} \]

subject to:

\[ \sum_{k \in I_{n}} Q_{k} - \sum_{k \in O_{n}} Q_{k} = D_{n} \]

\[ \Delta H_{k} = H_{n \in I_{k}} - H_{n \in O_{k}} \]

\[ \Delta H_{k} = w \left( \frac{Q_{k}}{C_{HW}} \right)^{\beta} L_{k} \sum_{m} d_{m}^{-\gamma} y_{km} \]

\[ H_{n} \geq H_{n}^{\text{min}} + E_{n} \]

\[ \sum_{m} y_{km} = 1 \]
Direct optimization

- Solve MINLP in GAMS, using DICOPT with the CPLEX MILP solver and a variety of NLP solvers:

<table>
<thead>
<tr>
<th>Initial Configuration</th>
<th>CONOPT2</th>
<th>CONOPT3</th>
<th>MINOS</th>
<th>MINOS5</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>659</td>
<td>655</td>
<td>444</td>
<td>Fails</td>
</tr>
<tr>
<td>All flows = 100</td>
<td>441</td>
<td>441</td>
<td>452</td>
<td>452</td>
</tr>
</tbody>
</table>

Initialization affects success of the NLP solvers.
- Large variation in quality of solutions obtained.
- Motivates development of an initialization procedure for robustness and consistency: hybrid approach
Visualization

- Computer based tools for design and optimization are intended for use by non-experts.
- Visual representations critical for ease of use.
- Interaction can enable engineer to apply own intuition.

- Interactive graphical display may provide initialization for rigorous optimization procedures.
Visualization

- Computer based tools for design and optimization are intended for use by non-experts.
- Visual representations critical for ease of use.
- Interaction can enable engineer to apply own intuition.

- Interactive graphical display may provide initialization for rigorous optimization procedures.
Discrete optimization

- Use of visualization requires mapping from continuous to discrete space.
- Mapping converts MINLP to discrete programming model ...
- ... but equality constraints cannot be satisfied in discrete space.
- So we use interval analysis to identify solutions which are close to feasible in discrete space.
- The discrete model is solved using a stochastic optimisation procedure.
Interval arithmetic

Changes to model given that node heads are now intervals:

\[ \Delta H_k = H_{n \in I_k} - H_{n \in O_k} \]

\[ Q_k = \left( \frac{\Delta H_k}{\frac{L_k}{C^\beta d_k^\gamma}} \right)^{1/\beta} \]

\[ 0 \in \sum_{k \in I_n} Q_k - \sum_{k \in O_n} Q_k - D_n \]

where \( \boxed{\text{[]}} \) indicates an interval value.
## Hybrid procedure results

<table>
<thead>
<tr>
<th>Initial Configuration</th>
<th>Solution ($10^3$ $\text{$}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CONOPT2</td>
</tr>
<tr>
<td>None</td>
<td>659</td>
</tr>
<tr>
<td>All flows = 100</td>
<td>441</td>
</tr>
<tr>
<td>Hybrid</td>
<td>419</td>
</tr>
</tbody>
</table>

- Behaviour of NLP solvers is more consistent.
- The global optimum is found in 3 of the cases.
- Solutions obtained are better in all cases.
Discrete to full MINLP

Correlation between solution obtained by mathematical programming versus approximate value of initial point generated by the genetic algorithm for Alperovits & Shamir example.

- Triangle points (△) indicate infeasible initial solutions.
- Crosses (×) infeasible final solutions.
Correlation between solution obtained by mathematical programming versus approximate value of initial point generated by the genetic algorithm for Hanoi example.
Discrete to full MINLP

Triangle points (△) indicate infeasible initial solutions.

Crosses (×) infeasible final solutions.

Correlation between solution obtained by mathematical programming versus approximate value of initial point generated by the genetic algorithm for Goulter & Morgan example.
Case Study 2: Heat exchanger networks

- Energy consumption is often the largest cost of a process.

- One means of reducing energy use is through process integration:
  - Identify matches for transfer of excess heat in one part of the process to another part.

- Problem is highly combinatorial, discontinuous and non-convex.
Graphical representations for integration

- Problem is defined by sets of hot and cold streams.
- Parts of streams are involved in matches.
- A stream may match with more than one other stream.
- Streams may split to match with different streams.

- A simple graphical technique is effective & intuitive...
- ... but not good for split streams.
Graphical representations for integration

- Problem is defined by sets of hot and cold streams.
- Parts of streams are involved in matches.
- A stream may match with more than one other stream.
- Streams may split to match with different streams.

- So we have developed a new ant colony model for stream splitting.
Starting point: HEN superstructure

The ant colony method provides us with a reduced superstructure:

which defines a nonlinear programming model.

Wish to extract solution from this.
Multi-start/Sequential algorithm

Let $\mathbf{x} \leftarrow 0$
converged $\leftarrow$ false

while not converged do
  $\mathbf{x}^{(1)} \leftarrow \mathbf{x}$
improved $\leftarrow$ true
  while improved do
    $\mathbf{x}^{(2)} \leftarrow \text{best solution from direct search methods applied to } \mathbf{x}^{(1)}$
improved $\leftarrow ||\mathbf{x}^{(2)} - \mathbf{x}^{(1)}|| > \epsilon$
    $\mathbf{x}^{(1)} \leftarrow \mathbf{x}^{(2)}$
  end while
  $\mathbf{x}^{(3)} \leftarrow \text{GA : initial population } = \{ \mathbf{x}^{(2)} \}$
  $\mathbf{x}^{(4)} \leftarrow \text{SA : } \mathbf{x}^{(3)}$
  $\mathbf{x}^{(5)} \leftarrow \text{engineer interaction!}$
  converged $\leftarrow ||\mathbf{x} - \mathbf{x}^{(5)}|| < \epsilon$
  $\mathbf{x} \leftarrow \mathbf{x}^{(5)}$
end while

$\triangleright$ Guaranteed feasible point

$\triangleright$ For direct search only

$\triangleright$ New iterate
## Results

<table>
<thead>
<tr>
<th>Variable</th>
<th>Step</th>
<th>Initial</th>
<th>IF</th>
<th>HJ</th>
<th>BFGS</th>
<th>GA+SA</th>
<th>UI</th>
<th>HJ</th>
<th>UI</th>
</tr>
</thead>
<tbody>
<tr>
<td>x11</td>
<td></td>
<td>0.0000</td>
<td>.807</td>
<td>.998</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>x12</td>
<td></td>
<td>0.0000</td>
<td>.697</td>
<td>.998</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>x13</td>
<td></td>
<td>0.0000</td>
<td>.695</td>
<td>.999</td>
<td>1.000</td>
<td>.998</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>x21</td>
<td></td>
<td>0.0000</td>
<td>.806</td>
<td>.744</td>
<td>.744</td>
<td>.745</td>
<td>.745</td>
<td>.745</td>
<td>.745</td>
</tr>
<tr>
<td>x31</td>
<td></td>
<td>0.0000</td>
<td>.962</td>
<td>.998</td>
<td>.999</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>x32</td>
<td></td>
<td>0.0000</td>
<td>.695</td>
<td>.999</td>
<td>.999</td>
<td>.320</td>
<td>.320</td>
<td>.320</td>
<td>.320</td>
</tr>
<tr>
<td>x33</td>
<td></td>
<td>0.0000</td>
<td>.694</td>
<td>.998</td>
<td>.998</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>sH11</td>
<td></td>
<td>0.0000</td>
<td>.688</td>
<td>.649</td>
<td>.649</td>
<td>.663</td>
<td>.663</td>
<td>.663</td>
<td>.663</td>
</tr>
<tr>
<td>sH12</td>
<td></td>
<td>0.0000</td>
<td>.193</td>
<td>.373</td>
<td>.374</td>
<td>.489</td>
<td>.489</td>
<td>.493</td>
<td>.493</td>
</tr>
<tr>
<td>sH21</td>
<td></td>
<td>0.0000</td>
<td>.695</td>
<td>.999</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>sH31</td>
<td></td>
<td>0.0000</td>
<td>.694</td>
<td>.999</td>
<td>.999</td>
<td>.039</td>
<td>.039</td>
<td>.004</td>
<td>0.00</td>
</tr>
<tr>
<td>sC12</td>
<td></td>
<td>0.0000</td>
<td>.693</td>
<td>.599</td>
<td>.600</td>
<td>.674</td>
<td>.674</td>
<td>.678</td>
<td>.678</td>
</tr>
<tr>
<td>sC11</td>
<td></td>
<td>0.0000</td>
<td>.693</td>
<td>.834</td>
<td>.834</td>
<td>.994</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

\[
f(x) \times 10^{-6}
\]

|        |      | 3.306  | 1.211 | .914 | .912 | .908 | .893 | .892 | .888 |

Hybrid methods for optimisation, INYS Workshop, Vilnius, 2006.02.14
Evolution of $f(x)$

[Graph showing the evolution of the objective function from initial to various steps including IF, HJ, BFGS, GA+SA, UI, HJ, UI.]
Evolution of $x_{32}$
Evolution of sH31

![Graph showing the evolution of sH31](image-url)
Evolution of $sH_{12}$
Summary

- Optimization problems in design are typically complex.
- Standalone single optimization techniques often not suitable alone.
- Combining methods into hybrid procedures can inherit the advantages of the individual methods.
- User interaction can also play a key part.
Acknowledgements

The following have contributed to the work presented here:

Dr Lazaros Papageorgiou, UCL
Dr Glenn Rowe, Dundee

and I would like to acknowledge the financial support of the British Council for attending this workshop.