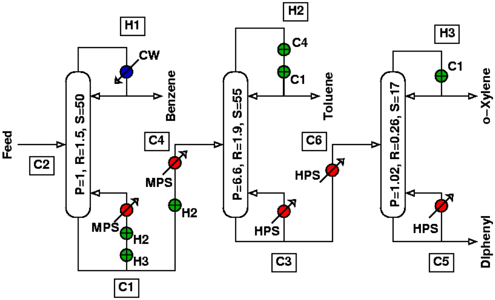
Evolutionary search for process design

Eric S Fraga
Centre for Process Systems Engineering
Department of Chemical Engineering
UCL

The 13th CREST Open Workshop
12-13 May 2011

Topic

Process design



- Identify process steps and their interconnections (mass, energy, control) to achieve performance goals.
- Wish to support decision making.
- Exploration necessary for understanding.
- Model based.

Optimisation

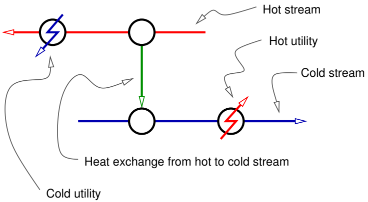
Need robust optimisation methods for process design:

- non-linear, non-convex, and discontinuous models,
- combinatorial search space,
- small, possibly non-convex, feasible regions, and
- ill- or un-defined objective function and constraint equations outside feasible regions.

Evolutionary methods can be a key element in the repertoire of optimisation tools an engineer may use.

Heat exchanger networks (HEN)

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.

Topic

1. Introduction
2. **Grammar**
3. Visual
4. Conclusions

Evolving a HEN structure

The key to any evolutionary algorithm is the encoding of alternative solutions (designs):

- We consider a biological analogue:

Genotype	the plan
Phenotype	the instance

- For HEN, the aim is to use a genotype to represent overall structure: possible matches and stream splits.
- The phenotype is an instantiation of the genotype with specific matches.

We use a Lindenmayer System (L-System) to represent and to evolve genotypes.

L-system definition

An L-system is defined by a tuple,

$$G = \langle V, \omega, P \rangle \tag{1}$$

where

V	the alphabet or set of symbols which can be replaced in a string by specific combinations symbols from the same set.
ω	the initial configuration (set of strings).
P	$(\subset V \times V^*)$ is the set of replacement rules.

For the heat exchanger network synthesis problem, we define an L-system, G_{HEN}.

ESF (2009), Engineering Optimization 41(9):813-831.

G_{HEN} – 1. Symbols V

The alphabet includes:

+, -	denote the cooling and heating requirements of each stream;
S, E	the start and end of each stream;
x	indication of exchange;
s, m	split and mix; and,
	start and end of split stream segments.
[,]	start and end of split stream segments

The full alphabet, therefore, is

$$V \equiv \{-,+,\mathtt{S},\mathtt{E},\mathtt{s},\mathtt{m},[,]\} \tag{2}$$

G_{HEN} – 2. Starting representation, ω

- The starting set of symbols is a set of strings, one for each stream in the network problem definition.
- Each cold stream is represented initially by the string `S-E`.
- Each hot stream by `E+S`.
- The hot and cold streams are written in opposite order to indicate the use of counter-current heat exchangers.

For our example, $\omega = \overbrace{\texttt{E+S}}^{\texttt{H1}}\ \overbrace{\texttt{E+S}}^{\texttt{H2}}\ \underbrace{\texttt{S-E}}_{\texttt{C1}}\ \underbrace{\texttt{S-E}}_{\texttt{C2}}.$

G_{HEN} – 3. Rule set, P

Rule	Target	Replacement	Description
R1	`-`	`x-`	Add an exchanger to a cold stream
R2	`-`	`s[x-][x-]m-`	Split a cold stream
R3	`+`	`x+`	Add an exchanger to a hot stream
R4	`+`	`m]x+[]x+[s+`	Split a hot stream
R5	`S`	`S`	A do-nothing rule

Note: the rule for splitting a hot stream creates a structure that is the reverse of that created by a cold stream split rule.

G_{HEN} summary

- The L-system for HEN design is context free.
- It is nondeterministic, complementing the evolutionary algorithm.
- The majority of strings (words generated from V^* starting with ω) represent a valid genotype for the HEN design problem.
- The genotype describes a configuration with
  - locations of splits and
  - locations for integrated exchangers.
- Genotype is instantiated into a phenotype for an actual design.

Phenotype instantiation

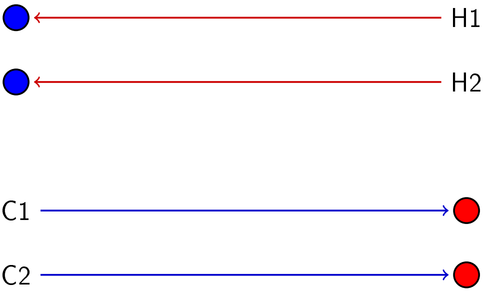
- A genotype describes the overall structure of a HEN.
- To instantiate a phenotype from a genotype:
  1. link the integrated exchangers and
  2. define the appropriate optimisation problem to size exchangers and determine split factors.
- Linking exchangers is non-deterministic so a single genotype may lead to different phenotypes.
- The do-nothing rule, R5, allows for multiple instances of the same genotype in a population.

Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`

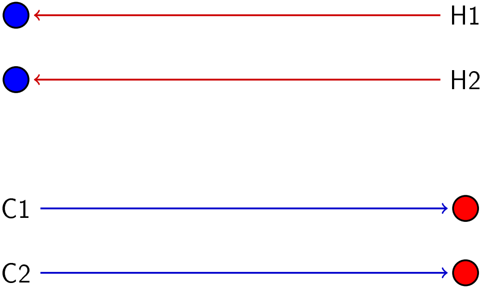
Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`



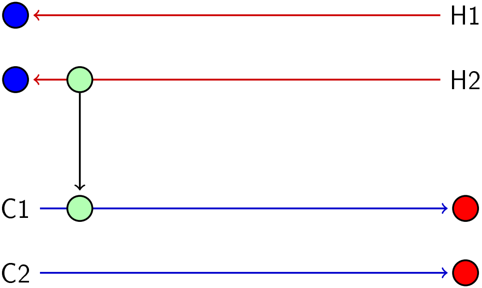
Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`
+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`



Example (re-visited)

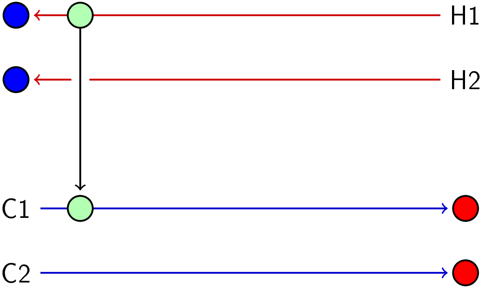
Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`
+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`



`H1:E+S, H2:Ex(C1)+S, C1:Sx(H2)-E, C2:S-E;`

Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`
+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`



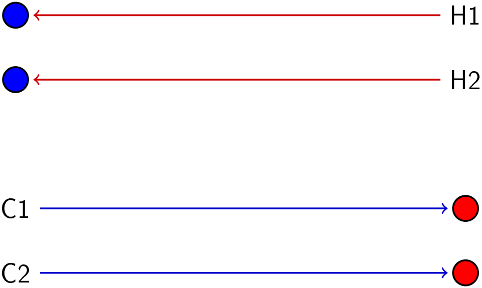
`H1:Ex(C1)+S, H2:E+S, C1:Sx(H2)-E, C2:S-E;`

Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`

+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`

+R2: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:Ss[x-][x-]m-E; }`

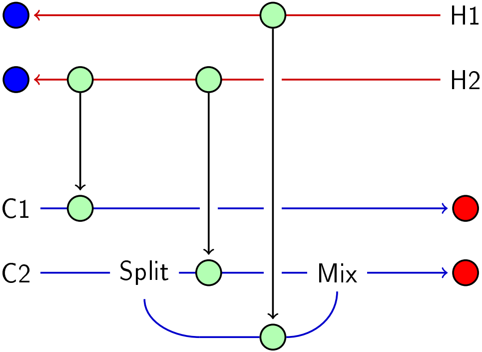


Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`

+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`

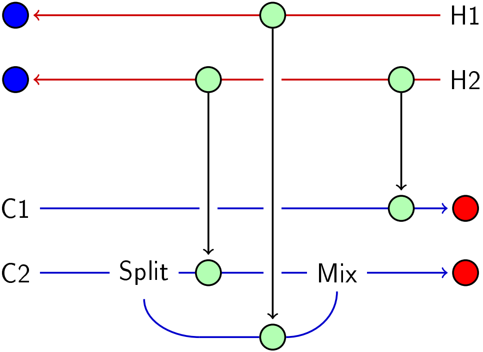
+R2: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:Ss[x-][x-]m-E; }`



`H1:Ex(C2)+S, H2:Ex(C1)x(C2)+S, C1:Sx(H2)-E, C2:Ss[x(H2)-][x(H1)-]m-E;`

Example (re-visited)

Initial: `{ H1:E+S, H2:E+S, C1:S-E, C2:S-E; }`
+R1: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:S-E; }`
+R2: `{ H1:E+S, H2:E+S, C1:Sx-E, C2:Ss[x-][x-]m-E; }`

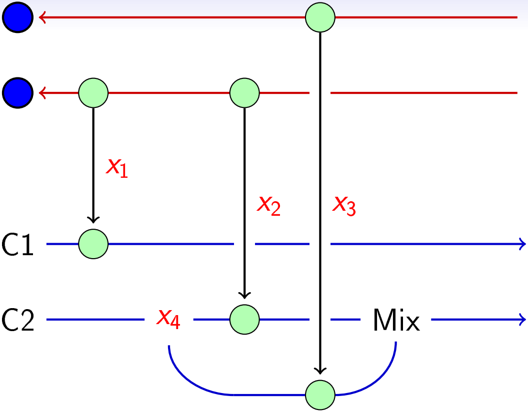


`H1:Ex(C2)+S, H2:Ex(C2)x(C1)+S, C1:Sx(H2)-E, C2:Ss[x(H2)-][x(H1)-]m-E;`

Embedded optimisation problem

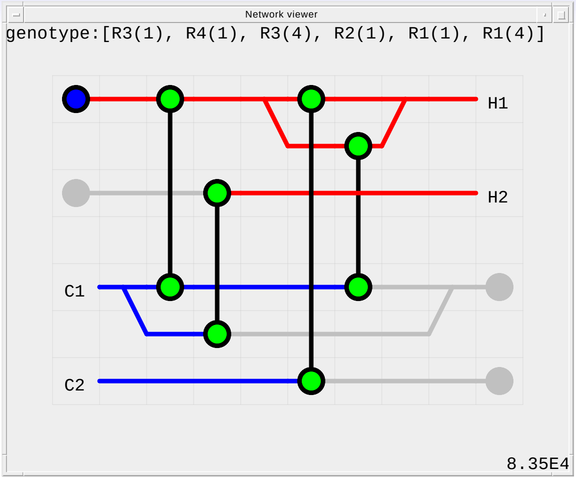
Phenotype ⇒ network structure ⇒ a nonlinear programme with decision variables:

1. $x_i \in [0,1]$, the fraction to exchange: $Q_i = x_i Q_{i,\max}$
2. $x_j \in [0,1]$, the split fraction.



The NLP represents a superstructure and is solved using a hybrid stochastic & direct search procedure.

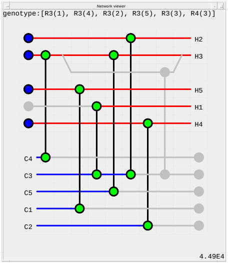
Example (solution)



genotype:[R3(1), R4(1), R3(4), R2(1), R1(1), R1(4)]

8.35E4

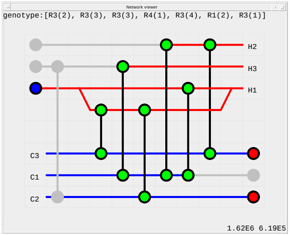
Other case studies



Network viewer

genotype:[R3(1), R3(4), R3(2), R3(5), R3(3), R4(3)]

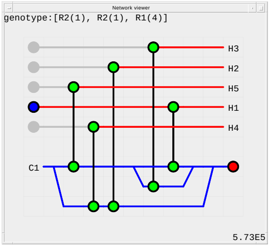
4.49E4



Network viewer

genotype:[R3(2), R3(3), R3(3), R4(1), R3(4), R1(2), R3(1)]

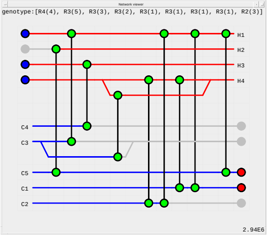
1.62E6 6.19E5



Network viewer

genotype:[R2(1), R2(1), R1(4)]

5.73E5



Network viewer

genotype:[R4(4), R3(5), R3(3), R3(2), R3(1), R3(1), R3(1), R3(1), R2(3)]

2.94E6

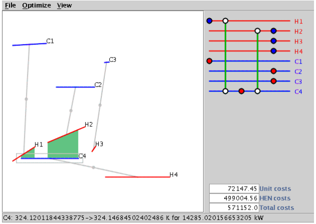
Topic

1. Introduction
2. Grammar
3. **Visual**
4. Conclusions

Visual representation

For a given process configuration, we can display the hot and cold streams visually and support interaction, where

- *x*-axis for position independent duties,
- *y*-axis for temperature, and
- hot stream overlapping cold stream indicates heat integration.

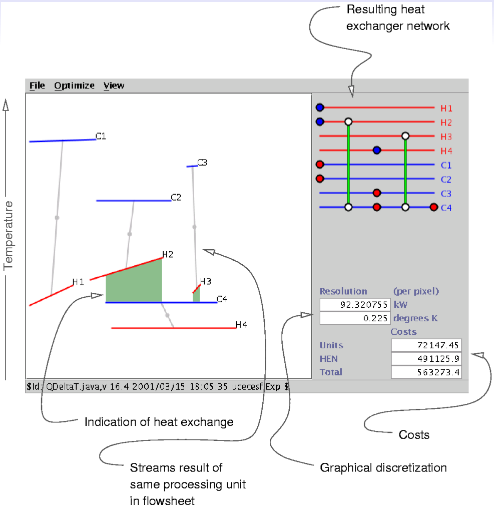


72147.45 Unit costs
499004.56 HEN costs
571152.0 Total costs

C4: 324.12011844338775->324.14684502402486 K for 14285.020156653205 kW

Allow user to manipulate process by moving streams (the tail wagging dog approach): streams can be moved horizontally for different integrations and moved vertically or stretched horizontally to change underlying unit designs.

Demonstration of interaction



Resolution (per pixel)
92.320755 kW
0.225 degrees K

Costs
Units 72147.45
HEN 491125.9
Total 563273.4

```
$Id: QDeltaT.java,v 16.4 2001/03/15 18:05:35 ucecesf Exp $
```

Resulting heat exchanger network

Indication of heat exchange

Costs

Streams result of same processing unit in flowsheet

Graphical discretization

- Interface encourages exploration and allows for insight.
- However, space is complex so best solutions are not always immediately obvious.

www

ESF, Patel & Rowe (2001). ChERD 79(7):765–776

Discrete design representation

Representation

A graphical view of process heat requirements defines left and right end-points for each hot and cold stream in the process:

$$\{(x_{a,i}, y_{a,i})\}$$

$$\{(x_{b,i}, y_{b,i})\}$$

$i = 1, \ldots, n_s$ and $x, y \in \mathbb{Z}^+$, suitable for manipulation by evolutionary algorithms.

1. Define list of intervals

$$I \leftarrow \bigcup_{1}^{n_s} \{\{x_{a,i}\} \cup \{x_{b,i}\}\}$$

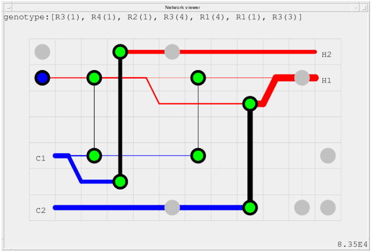
2. For each interval $[I_j, I_{j+1}]$:
   1. Generate list of active streams, A.
   2. Sort A from top to bottom using y_b values.
   3. Generate match for each hot stream immediately above cold stream in A.
   4. Generate utility match for all other streams.
3. Coalesce adjacent similar matches.
4. Design exchanger for each match.
5. Cost all exchangers and utility use.

Topic

1. Introduction
2. Grammar
3. Visual
4. Conclusions

Summary

- Optimization problems in design are typically complex.
- Evolutionary methods can provide a useful tool for design.
- The key is the representation of possible solutions.



Network viewer

```
genotype:[R3(1), R4(1), R2(1), R3(4), R1(4), R1(1), R3(3)]
```

Acknowledgements

- Ms Rupal Patel, UCL
- Dr Glenn Rowe, University of Dundee
- and the attendees at the ACDM conference series (organiser: Professor Ian Parmee)

`http://www.homepages.ucl.ac.uk/~ucecesf/`