Heat exchanger network design

Mathematical programming approaches

Prof. François Marechal

Calculating of the optimal HEX areas

NLP(Non Linear Programming) optimisation problem

$$\min_{\dot{M}_u, A_e} = \left(\sum_{u=1}^{N_u} c_u^+ \cdot \dot{M}_u\right) \cdot t_{op} + \frac{1}{\tau} \sum_{e=1}^{n_{htx}} \left(a_e + b_e \cdot (A_e)^{c_e}\right)$$

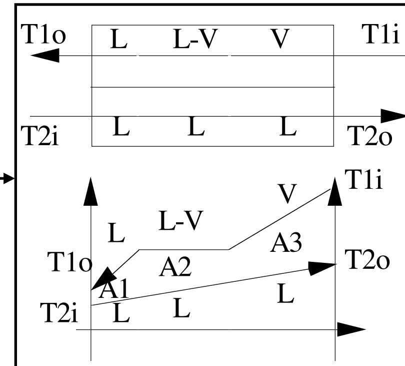
Constraints

Heat and mass balances
Rating equations
Specifications :

F(X) = 0

Bounds and limits

G(X) ≤ 0



$$A_i = \sum_{z=1}^{nz_i} A_{z,i} = \sum_{z=1}^{nz_i} \frac{Q_{iz}}{U_{iz} \Delta Tlm_{iz}}$$

X : flows (split factors, utility flows), pressure, temperature, area, heat exchanged, ...

Optimising the flows (e.g. in splitters and utility streams) and temperatures therefore changing the heat exchange areas allows to calculate the best ΔTmin value for each of the heat exchangers (EMAT). The calculation is a non linear programming problem that can be quite complex to solve, due to the interrelations between the heat exchangers and the difficulty of the infeasible heat exchanges (ΔT < 0).
Non linear programming : class of optimisation that involves non linear equality (heat balances, heat transfer) and inequality constraints (flows >0, ΔT>0).

Difficulty of the method

- Sequential approach
  - interdependent decisions
- Multiple solutions
- Do we reach the minimum number of units ?
- Is the network optimal ?

- Reducing the number of units
  - save the (fixed part) of investment
  - first m2 costs more
- Optimisation is needed
  - $EMAT_i$ optimisation

Application of mathematical programming

- **Calculating heat exchanger network design**
    - given a list of all the hot and cold streams
    - given the ΔTmin and the pinch(s) location
  - Calculate :
    - the heat exchanger network structure
    - The heat loads, temperature and flows in the network
    - The heat exchange area of the heat exchangers
    - The total cost of the heat exchanger network
    - The optimal value of ΔTmin for each HEN
- Mathematical programming (optimisation) can help identifying the matches between the hot and the cold streams and therefore reduce the risk of taking a wrong decision in the sequential approach.

EPFL

HEN synthesis

Draw backs of the Pinch Design Method

- multiple solutions
- combinatorial problem
- sequential

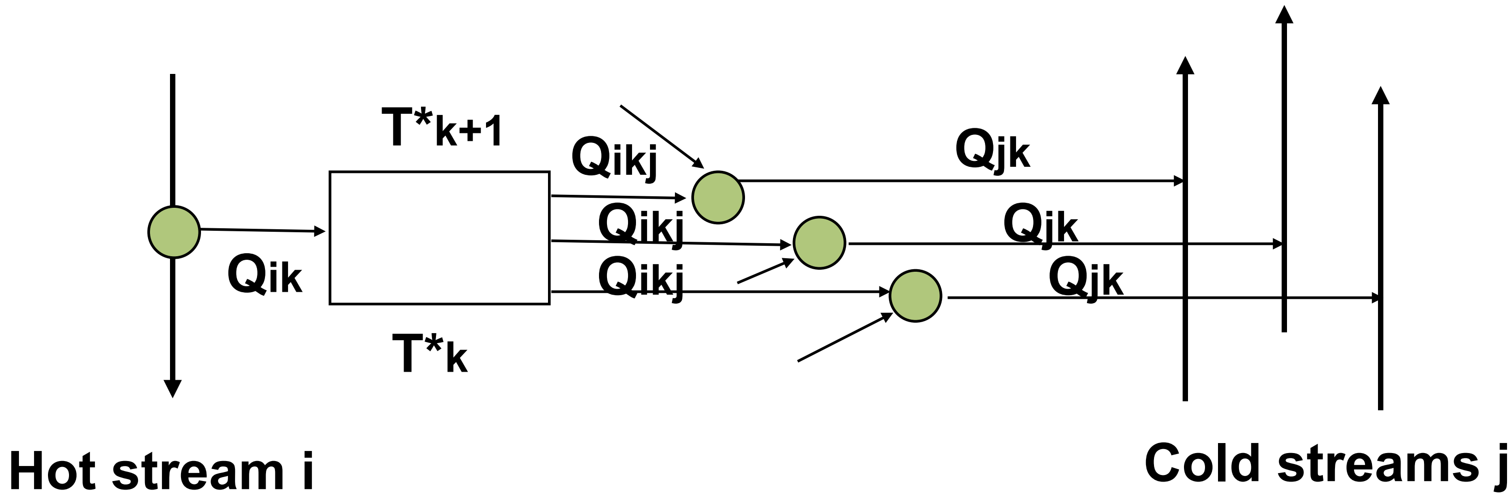
Use of mathematical programming:

Heat load distribution:

- which streams exchange heat
- How much heat is exchanged
- minimize the number of connections
- satisfies DTmin and MER

Remaining problem : find the HEN structure

Heat load distribution



<u>Hot stream i in temperature interval k</u>

$$\sum_{j=1}^{nc} Q_{ikj} = Q_{ik} \qquad \forall i = 1,\ldots,nh; \forall k = k_1,\ldots,k_2$$

<u>Cold stream j in and above temperature interval k</u>

$$\sum_{i=1}^{nh}\sum_{r=k}^{k_2} Q_{irj} - \sum_{r=k}^{k_2} Q_{jr} \leq 0 \qquad \forall j = 1,\ldots,nc; \forall k = k_1,\ldots,k_2$$

<u>connection between i et j (integer variable)</u>

$$\sum_{r=k}^{k_2} Q_{irj} - y_{ij} Q\max_{ij} \leq 0 \qquad \forall j = 1,\ldots,nc; \forall i = 1,\ldots,nh$$

Q_{ikj} the heat load of the hot stream i in temperature interval k that is sent to the cold stream j
Q_{ik} the heat load of hot stream i in temperature interval k
Q_{jk} the heat load of cold stream j in the temperature interval k

Heat load distribution

MILP formulation (Mixed Integer Linear Porgamming)

Minimize the number of connections

$$\underset{y_{ij},Q_{ikj}}{Min} \sum_{i=1}^{nh} \sum_{j=1}^{nc} y_{ij} \qquad y_{ij} \in \{0,1\}$$

$$\sum_{j=1}^{nc} Q_{ikj} = Q_{ik} \qquad \forall i = 1,\ldots,nh; \forall k = k_1,\ldots,k_2$$

$$\sum_{i=1}^{nh} \sum_{r=k}^{k_2} Q_{irj} - \sum_{r=k}^{k_2} Q_{jr} \leq 0 \qquad \forall j = 1,\ldots,nc; \forall k = k_1,\ldots,k_2$$

$$\sum_{r=k}^{k_2} Q_{irj} - y_{ij} Q\max_{ij} \leq 0 \qquad \forall j = 1,\ldots,nc; \forall i = 1,\ldots,nh$$

The problem is a mixed integer linear programming problem (MILP). This is class of optimisation problem that has only linear constraints and objective function and in which the decision variables are continuous (Q_{ikj}) or integer (y_{ij}).

Multiple solutions

- Add heuristic rules
  - favour the connexion with utility streams
  - favour close connexions
  - favour connexion in closer sub-systems
- A heuristic rule is applied only if it does not penalize the minimum number of solution target

$$\min_{y_{i,j},Q_{i,j,k}} \sum_{i=1}^{N_h} \sum_{j=1}^{N_c} W_{i,j} \cdot y_{i,j}$$

$$W_{i,j}\text{: weighting factor of connexion i,j}$$

There exist most of the time more than one solution that minimises the number of connections. In order to distinguish the solutions it is therefore important to give an importance to some of the solutions. This can be done by preferences or by other weighting criteria (e.g. the geographical distance that would indicate the pimping required to connect two streams).

Introduce heuristic rules in MILP programs

- **The weight of priority rule k is given by :**
  - the number of possible connexions satisfying rule k

$$P_k = \sum_{j=1}^{n_c} \sum_{i=1}^{n_h} (p_{kij})$$

  - an im

$$\min_{y_{ij} Q_{ikj}} NT = \sum_{j=1}^{n_c} \sum_{i=1}^{n_h} \left(\prod_{k=1}^{p_{ij}-1} (P_k + 1) \, y_{ij} \right)$$

$$\frac{\prod_{k=1}^{r} (P_k + 1)}{\prod_{k=1}^{r-1} (P_k + 1)} = P_r + 1 > \sum_{i=1}^{n_h} \sum_{j=1}^{n_c} P_{rij}$$

When we can define a order of preferences between the solutions, the above formulation guarantees that the number of connections will be minimized (i.e. the sum of the previous preference will never exceed the value of the next preference level with one more unit)

Generating multiple solutions

- Integer cut constraint : generate the next best solution
  - assuming that we know k solutions
  - problem k + 1 is defined by adding to the previous MILP problem the integer cut constraint

$$Problem^{k+1} =$$
$$Problem^{k} +$$
$$\sum_{p=1}^{n_p}\sum_{c=1}^{n_c}(2 * y^k_{p,c} - 1) * y_{p,c} \leqslant \sum_{p=1}^{n_p}\sum_{c=1}^{n_c} y^k_{p,c}$$

The integer cut contraints indicates as a constraints the list of solutions that are already known. The solution to this problem is therefore the next best solution.

Conclusion

- **HEN design algorithm**
  - sequential approach
  - thermo-economic evaluation of options
- **HEN design with mathematical programming**
  - systematic with a holistic approach
  - decides the connections with mathematical programming approach
  - HEN network still needs to be defined
    - Split streams
    - Non linear programming