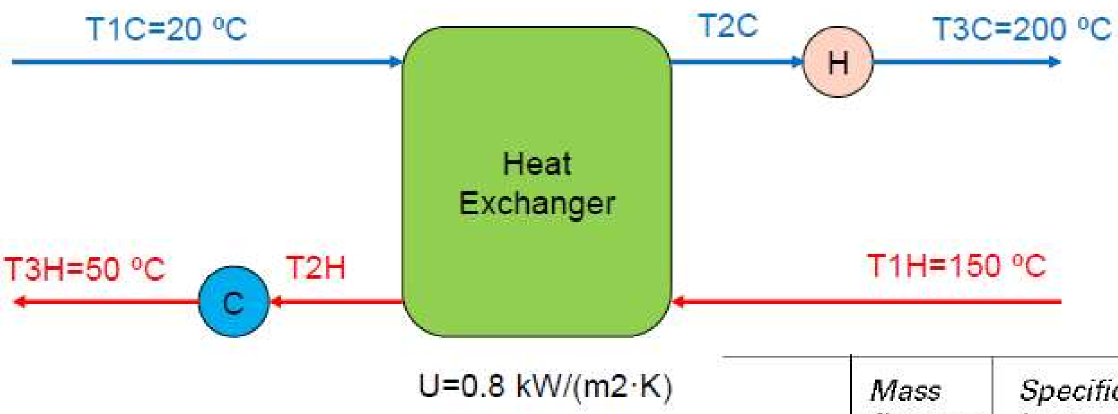
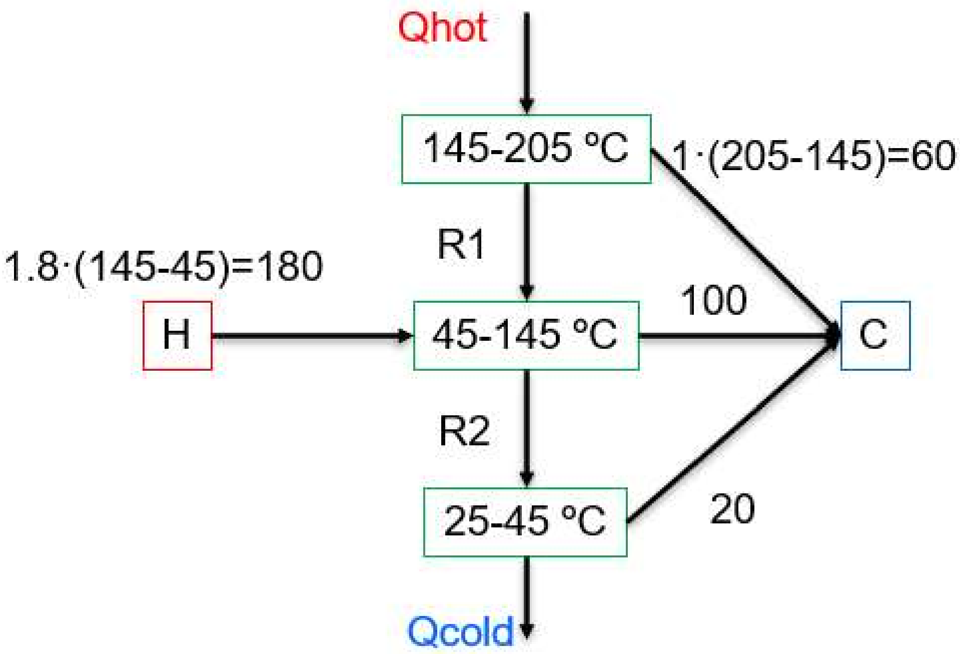
Se desea maximizar el caudal de calor intercambiado del siguiente sistema, para lo cual se dispone de dos variables de decisión (T2C y T2H)



	Mass flowrate W (kg/s)	*Specific heat capacity C_P (kJ/kgK)*
Cold stream	0.25	4
Hot stream	0.4	4.5

En este caso, resolveremos el ejemplo utilizando el "Transhipment model", en el cual se plantean los balances alrededor de cada intervalo de temperaturas



Recordad que las temperaturas están desplazadas E.g. En el caso de T1H=150, al ser una temperatura de corriente caliente disminuye en una cantidad $\Delta T_m/2$. Para el caso de T3C=200 pasa a ser 205 C al ser una corriente fría.

In [ ]:
```
from pyomo.environ import *
model = ConcreteModel()
```

Variables

In [ ]:
```
Qhot = model.Qhot = Var(within = NonNegativeReals)
Qcold = model.Qcold = Var(within = NonNegativeReals)
R1 = model.R1 = Var(within = NonNegativeReals)
R2 = model.R2 = Var(within = NonNegativeReals)
```

Objective function

In [ ]:
```
model.util = Objective(expr = Qhot + Qcold)
```

Constraints

In [ ]:
```
model.int1 = Constraint(expr = Qhot - 60 - R1 == 0)
model.int2 = Constraint(expr = 180 + R1 - R2 - 100 == 0)
model.int3 = Constraint(expr = R2 - 20 - Qcold == 0)
```

Solution

In [ ]:
```
results = SolverFactory('glpk').solve(model)
model.pprint()
results.write()
```

```
4 Var Declarations
    Qcold : Size=1, Index=None
        Key  : Lower : Value : Upper : Fixed : Stale : Domain
        None :     0 :  60.0 :  None : False : False : NonNegativeReals
    Qhot : Size=1, Index=None
        Key  : Lower : Value : Upper : Fixed : Stale : Domain
        None :     0 :  60.0 :  None : False : False : NonNegativeReals
    R1 : Size=1, Index=None
        Key  : Lower : Value : Upper : Fixed : Stale : Domain
        None :     0 :   0.0 :  None : False : False : NonNegativeReals
    R2 : Size=1, Index=None
        Key  : Lower : Value : Upper : Fixed : Stale : Domain
        None :     0 :  80.0 :  None : False : False : NonNegativeReals

1 Objective Declarations
    util : Size=1, Index=None, Active=True
        Key  : Active : Sense    : Expression
        None :   True : minimize : Qhot + Qcold

3 Constraint Declarations
    int1 : Size=1, Index=None, Active=True
        Key  : Lower : Body            : Upper : Active
        None :   0.0 : Qhot - 60 - R1 :   0.0 :   True
    int2 : Size=1, Index=None, Active=True
        Key  : Lower : Body                  : Upper : Active
        None :   0.0 : 180 + R1 - R2 - 100 :   0.0 :   True
    int3 : Size=1, Index=None, Active=True
        Key  : Lower : Body             : Upper : Active
        None :   0.0 : R2 - 20 - Qcold :   0.0 :   True

8 Declarations: Qhot Qcold R1 R2 util int1 int2 int3
# ==========================================================
# = Solver Results                                         =
# ==========================================================
# ----------------------------------------------------------
#   Problem Information
```

```
# ----------------------------------------------------------
Problem:
- Name: unknown
  Lower bound: 120.0
  Upper bound: 120.0
  Number of objectives: 1
  Number of constraints: 4
  Number of variables: 5
  Number of nonzeros: 7
  Sense: minimize
# ----------------------------------------------------------
#   Solver Information
# ----------------------------------------------------------
Solver:
- Status: ok
  Termination condition: optimal
  Statistics:
    Branch and bound:
      Number of bounded subproblems: 0
      Number of created subproblems: 0
  Error rc: 0
  Time: 0.057425498962402344
# ----------------------------------------------------------
#   Solution Information
# ----------------------------------------------------------
Solution:
- number of solutions: 0
  number of solutions displayed: 0
```

In [ ]:

```python
Qc = value(model.Qcold)
Qh = value(model.Qhot)
print('Cold utility = {0:2.2f}, Hot utility = {1:2.2f}'.format(Qc, Qh))
```

```
Cold utility = 60.00, Hot utility = 60.00
```