

Part 3A

Mathematical Programming Transshipment Model

数学规划与运输模型

引言 Introduction

回顾一下 Part 2 的夹点技术 (Pinch Technology)：

- ① 运用夹点技术设计换热网络需要两步：先确定公用工程量，然后再设计
- ② 在最优化的时候使用的是近似的换热面积和最小传热单元数，而换热单元数对 cost 的影响更大
- ③ 所有的换热器都遵循同一最小传热温差
- ④ There is no shell counting
- ⑤ 没有理论（或者说固定的步骤）处理远离夹点处的分流
- ⑥ 不系统：在完成夹点附近的物流匹配 (pinch match) 之后，就只剩下需要直觉来构建专业知识了（意思就是剩下的物流匹配就只能发挥主观能动性了）
- ⑦ 问题规模越大，设计难度越大
- ⑧ 在估计换热面积的时候，我们假设的是垂直换热。实际换热网络里，在夹点附近确实是垂直换热，但随着区域远离夹点，为了减小传热单元数，换热就不再是垂直换热了。
- ⑨ 允许跨越夹点传热有些时候会更有优势

所以，夹点技术存在一定的问题。那有没有别的方法，可以做到：

- ① 通过衡量成本（能量、面积、设备）来同时处理所有的权衡。摒弃两步设计与能量放松 (relaxation)，让真正的经济因素评判设计。
- ② 移除 HRAT 的限制，并在适当或需要的时候对特定的物流对引入换热器最低近似温度 (exchanger minimum approximation temperatures, EMAT)
- ③ 不表现出如缺少对分流的控制或者等温混合这样的限制
- ④ 系统的，可能是自动的

这意味着我们需要引入高效的方法与工具。我们要解决的是诸多限制下的最优化问题，也就是数学规划问题。计算机程序是一定要引入的。有这么一个软件可以帮你求解这类问题，它就是 GAMS。

二、GAMS 使用说明

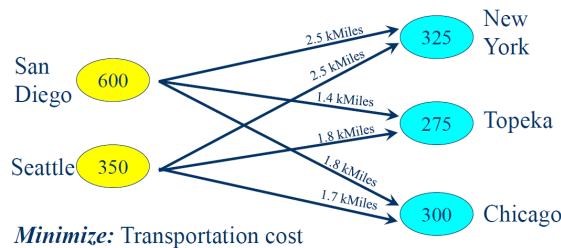
关于 GAMS 的具体语法，可以通过附件中的手册了解。这门课不要求会写，只要能够看懂后面化工系统工程相关的问题的代码即可。

什么是 GAMS？它是通用代数建模系统 General Algebraic Modeling System 的简称，这个系统用于解决线性、非线性和混合整数优化问题的建模。非常适合规模大且复杂的问题

GAMS 的界面有点像计算机程序的 IDE，也就是说，你需要敲代码。但和 C、Java、Python 这些语言不同，GAMS 的代码是用来描述问题而不是解决问题的。当你的问题以规范的形式交给程序后，程序会通过内置的代码帮你求解。

接下来用一个例子简单讲述一段完整的 GAMS 代码是怎么构成的。

问题：假设有 2 个供应地、3 个市场、1 种货品，已知每千英里的运输成本为 \$90。为了使总运输成本最小，应该如何分配每条路线的运输量？



首先先进行数学建模，将它转化为一个纯数学问题：

问题的自变量是每个路径的分配量

问题的约束条件是供应地和接收地的供应、接收限制：

Distances				
	Markets			
Plants	New York	Chicago	Topeka	Supply
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
Demand	325	300	275	

问题的目标是求总成本的最小值，那么：

设 i — 供应地， j — 需求地

（不要将 i 和 j 理解为整数，请理解为“供应地”和“需求地”两个集合的元素）

a_i — 供应地 i 能够提供的商品数量

b_j — 需求地 j 需要的商品数量

c_{ij} — 从供应地 i 到需求地 j 的单位运货量成本

x_{ij} — 从供应地 i 到需求地 j 的运货量，大于等于 0

限制条件：① 对每一个 i ， $\sum_{j=1}^J x_{ij} \leq a_i$ ；② 对每一个 j ， $\sum_{i=1}^I x_{ij} = b_j$

目标：确定所有的 x_{ij} ，使 $z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$ 取到最小值

这就是这个问题的模型了，在这个问题中，我们定义了很多量，它们都是相同的吗？

i 和 j 是下标，但不是整数 1、2、3……而是代表了某个集合中的元素

a_i , b_j , c_{ij} 都是和 i 和 j 有关的量，好比是一个集合的函数，在求解给定的问题时是不变量

x_{ij} 、 z 也是和 i 和 j 有关的量，集合的函数，在求解给定的问题时是变化量

此外，限制条件和目标也是和上面这些量不同但又类似的东西

所以，GAMS 中，一个数学规划问题模型中的所有内容被划分为 6 个类型，我们通过对上面这个问题的代码讲解来介绍这 6 个类型

- ① 和 C 语言一样，每一条语句都要以 “;” 结尾
- ② GAMS 的编译器不区分大小写字母，务必注意

```
Sets
  i   canning plants   / seattle, san-diego /
  j   markets          / new-york, chicago, topeka / ;
```

Set 是第一种类型，这一段的作用是声明并命名了 2 个 Set 类型 i 和 j

i 和 j 后面没有加粗的蓝色文本为注释，让人能够看懂 i 和 j 是什么意思

Set 相当于一个集合，集合里面一定有元素，后面的绿色斜杠内就是集合的元素

它也有点像正经程序语言里数组的下标，不过数组的下标都是自然数，这里的“下标”可以是任何东西

上面这段是将两个 set 合并在一个句子里，所以语句的开头是 **Sets**。你也可以分开写，这样需要两个 **Set**：

```
Set i canning plants / Seattle, San-Diego / ;
Set j markets / New-York, Chicago, Topeka / ;
```

如果元素名字除了数字以外的部分相同，数字部分是一串连续的自然数，你就可以简写为：

```
Set t time periods /1991*2000/ ;
Set m machines /mach1*mach24/ ;
```

很多人搞不清 Parameter 和 set 的区别。Parameter 相当于定义了一个与 Set 有关的参数集合，这个集合中的元素和 Set 集合中的元素是一一对应的关系。只有 Parameter 才能赋值，Set 没有值。

```
Parameters

  a(i)  capacity of plant i in cases
        /   seattle      350
          san-diego     600 /

  b(j)  demand at market j in cases
        /   new-york     325
          chicago        300
          topeka         275 / ;
```

GAMS 有三种格式的数据类型，Parameter 就是其中一种，它类似于二维数组。

这一段了声明 Parameter 的命名以及它们各自的域。比如第一句中参数为 a ，域为集合 i ，意思就是 $a(i)$ 是在集合 i 上定义的函数。

后面的绿色部分就是为集合中的每个元素赋对应的参数值。每个“元素-值”配对必须用 “,” 分隔或换行分开

```
Table d(i,j)  distance in thousands of miles

           new-york      chicago      topeka
seattle      2.5          1.7          1.8
san-diego     2.5          1.8          1.4 ;
```

Table 就像二维数组，是在集合 i 和 j 组成的“矩阵”上定义的函数。

```
Scalar f freight in dollars per case per thousand miles /90/ ;
```

Scalar 就相当于单个变量了，它与 i 和 j 都没有对应的关系。

除了一开始就已知值的变量外，还有一些变量是后续计算出来的，它们都属于上面三种变量中的一种，可以只声明，不赋值

```
Parameter c(i,j) transport cost in thousands of dollars per case ;  
  
c(i,j) = f * d(i,j) / 1000 ;
```

参数运算的时候，语法类似 C 语言，除了幂运算直接用 “**” 表示外，其余和 C 没什么差别

除了运算后赋值外，也可以直接赋值，赋值会直接覆盖参数先前的值。

```
c('Seattle','New-York') = 0.40 ;
```

```
Variables  
x(i,j) shipment quantities in cases  
z total transportation costs in thousands of dollars ;  
  
Positive Variable x ;
```

Variables 就是在程序求解时会变动的量，分为自变量和因变量。

它们的类型有 FREE, POSITIVE, NEGATIVE, BINARY, or INTEGER。

如果后面不做声明，默认值就是 FREE

```
x.up(i,j) = capacity(i,j) ;  
x.lo(i,j) = 10.0 ;  
x.up('seattle','new-york') = 1.2*capacity(seattle,'new-york') ;
```

x.up 规定了自变量 x 的上限，x.lo 规定了自变量 x 的下限

```
Equations  
cost define objective function  
supply(i) observe supply limit at plant i  
demand(j) satisfy demand at market j ;  
  
cost .. z =e= sum({i,j}, c(i,j)*x(i,j)) ;  
  
supply(i) .. sum(j, x(i,j)) =l= a(i) ;  
  
demand(j) .. sum(i, x(i,j)) =g= b(j) ;
```

=e= 为 等于
=l= 为 小于等于
=g= 为 大于等于

声明类型的时候，如果要声明多个，那么开头的单词就要变成复数形式，这几个单词都很正常，加 s 就行。

Equation 是与 variable 有关的表达式。它们往往代表了问题的限制条件和目标函数

这一段声明并命名了 3 条 Equation，然后在下面给出了它们的表达式。

很多人也搞不清 Equation 和前面那些没有 Equation 开头的表达式的区别。

Equation 的表达式中一定包含至少一个 Variable，而那些没有 Equation 开头的表达式只含有 Parameter、Table 这类参数。从目的的角度区分，Equation 是给出问题的限制条件和目标函数，没有 Equation 开头的表达式是为了计算某个参数。

```
Model transport /all/;
```

Model 代表了一个问题，它的命名是独一无二的。

声明一个 Model 需要指定它包含哪些 Equation (本例中是“all”所有 Equation, 等价于 Model transport /cost, supply, demand/;)

Model 使得一个程序里可以用相同或不同的数据做不同的规划, 我们想要解决的问题往往都是一个大问题里面的小问题。

```
Solve transport using lp minimizing z ;
```

Solve 就是告诉 GAMS 程序你需要求解哪个模型, 并选择使用哪个求解器 (本例为 LP)、并指明最优化方向 (最小值 MINIMIZING 还是最大值 MAXIMIZING) 和最优化的变量 variable

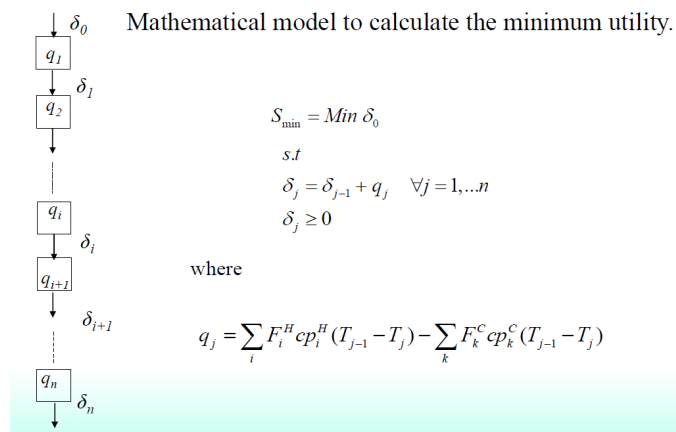
```
Display x.l, z.l ;
```

Display 就是让程序输出结果, 不一定是求解后的 variable, 求解前的参数也可以输出

三、GAMS 应用于问题表法

现在我们回到化工系统工程中来, 试着用 GAMS 解决一个问题

比如, 用问题表法求解最小 UTILITY



```
Sets i hotstreams /H1 * H2/,  
      j coldstreams /C1 * C2/;
```

Parameters

HRAT	最小传热温差
fh(i)	heat capacity flowrate of hot stream 热物流的 CP
fc(j)	heat capacity flowrate of cold stream 冷物流的 CP
thin(i)	supply temp. of hot stream 热物流进口温度
thout(i)	target temp. of hot stream 热物流出口温度
tcin(j)	supply temp. of cold stream 冷物流进口温度
tcout(j)	target temp. of cold stream; 冷物流出口温度

```
thin('H1') = 250; thout('H1') = 40; fh('H1') = 0.15;  
thin('H2') = 200; thout('H2') = 80; fh('H2') = 0.25;  
tcin('C1') = 20; tcout('C1') = 180; fc('C1') = 0.2;  
tcin('C2') = 140; tcout('C2') = 230; fc('C2') = 0.3;  
HRAT = 10 ;
```

Parameters

gamma(i,j) upper bound of driving force 两股物流之间的最大传热推动力
ech(i) heat content hot stream i 热物流 i 能够给出的热量
ecc(j) heat content stream cold j; 冷物流 j 能够吸收的热量

ech(i) = fh(i)*(thin(i) - thout(i)) ; $H_{\text{hot}} = CP \times (T_{\text{in}} - T_{\text{out}})$
ecc(j) = fc(j)*(tcout(j) - tcin(j)) ; $H_{\text{cold}} = CP \times (T_{\text{out}} - T_{\text{in}})$
gamma(i,j) = 1.01 * max(0,tcin(j) - thin(i), tcin(j) - thout(i), tcout(j) -
thin(i), tcout(j) - thout(i)) ;

下面正式开始计算最小 UTILITY

Set int temperature-interval /int1 * int10/; 定义温度区间的集合
这个 int 是 interval 的简写，不是 integer!!!

alias (int, ints); 为集合 int 再进行一次命名 ints，相当于复制一个集合

Parameter TINT(int); 表示温度区间，相邻两个参数值就是温度区间的上下界

Parameter TINTs(ints); 和上面那个一样

Parameter xnumber; 用于交换的临时变量

Scalars Hi, Cj ,Tintint, Tintints; 用于指示位置的临时变量

Parameter gg(int); 我也想知道这个是什么

Parameter qint (int); 温度区间内的净热量

Parameter qinth(int); 温度区间内热物流放出的热量

Parameter qintc(int); 温度区间内冷物流吸收的热量

Free Variable UMIN; 定义最小热公用工程量

Positive Variable delta0; 定义最开始持有的热量

Positive Variable delta(int); 定义经过各个温度区间后热量的盈余

Equation intervalbalance0;

intervalbalance0 .. delta('int1') =e= delta0 + qint('int1');

Equation intervalbalance;

intervalbalance(int)\$ (ord(int) ne 1) .. delta(int) =e= delta(int-1) +
qint(int);

ord(int)表示当前的这个 int 在集合中的次序

\$是条件符号，只有\$()里的语句成立，方程才会被执行

ne 表示不等于

这两条方程的意思就是计算 $\delta_i = \delta_{i-1} + q_i$

因为 int 集合从 int1 开始，不包含 int0，所以需要额外加一句 δ_0 如何计算

Equation minutil;

minutil.. UMIN =e= delta0;

规定了我们想要得到的 Umin 如何求得

Model Minimumutility /intervalbalance0, intervalbalance, minutil/;

建立求最小公用工程量的模型，包含了上面的方程（条件）

Parameter pinch_temmpature;

Tintints = 1;

FOR(Hi = 1 TO CARD(i), 函数 CARD()给出集合里元素的个数

TINTs(int)\$[ord(int) = Tintints] = sum(i\$[ord(i) = Hi],thin(i));

遍历集合 int 所有的元素，

如果 当前元素（温度区间）的次序 等于 Tintints 的值

那么 执行 TINTs(int)的赋值语句

遍历集合 i 所有的元素，

如果 当前元素的次序为 Hi 的值

那么 将该元素的 thin()参数计入 sum 的加和

遍历完成后，将 sum 的值赋给 TINTs(int)

总结：Tintints 的值是几代表着当前指向第几个区间

Hi 的值是几代表着当前指向第几个热物流

一句话：把 集合 i 中第 Hi 个元素的 thin 值 赋给 集合 int 中第 Tintints 个元素的 TINTs 值

用人话就是：依次将所有热物流的进口温度放进温度区间集合对应的参数里

Tintints = Tintints + 1;) ;

display TINTs;

FOR(Hi = 1 TO CARD(i),

TINTs(int)\$[ord(int)=Tintints]=sum(i\$[ord(i)=Hi],thout(i));

Tintints = Tintints + 1;);

display TINTs;

同样这里是依次将所有热物流的出口温度放进温度区间集合对应的参数里

FOR(Cj = 1 TO CARD(j),

TINTs(int)\$[ord(int)=Tintints]=sum(j\$[ord(j)=Cj],tcin(j)+Hrat);

Tintints=Tintints+1;);

display TINTs;

同样这里是依次将所有冷物流的进口温度放进温度区间集合对应的参数里

FOR(Cj=1 TO CARD(j),

TINTs(int)\$[ord(int)=Tintints]=sum(j\$[ord(j)=Cj],tcout(j)+Hrat);

Tintints=Tintints+1;);

display TINTs;

同样这里是依次将所有冷物流的出口温度放进温度区间集合对应的参数里

最后，得到了集合 int 的一个参数 TINTs(), 表示各个温度区间的上限或下限

下面是对我们得到的 TINTs()参数数组进行从小到大进行排序

如果你还记得大一学过的程序设计，能否看出这是哪个排序算法？

For(Hi = 1 to card (int),

For(Cj = Hi + 1 to card (int),

if(sum(int\$(ord(int) eq Hi),TINTs(int)) <= sum(int\$(ord(int) eq Cj),TINTs(int)),

xnumber = sum(int\$(ord(int) eq Hi),TINTs(int));

```

TINTs(int) $(ord(int) eq Hi) = sum(ints$(ord(ints) eq Cj),TINTs(ints));
TINTs(ints)$(ord(ints) eq Cj) = xnumber;
);
);
);

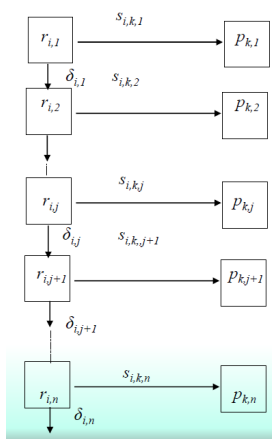
display TINTs;
TINT(int) = TINTs(int);    复制排完序的结果给 TINT()参数 (虽然我看不懂这个操作)
display TINT;
接下来计算各温度区间的热量盈余
FOR(Tintint = 1 TO (CARD(int)-1),    从小到大遍历温度区间
qinth(int)$[ord(int)=Tintint] = sum(i$[thin(i)>= TINT(int) AND thout(i) <=
TINT(int+1)], fh(i)*(TINT(int)-TINT(int+1)));
    在第 Tintint 个温度区间里, 遍历所有热物流, 如果一个物流的进口温度大于等于其温度区间
    的上界, 出口温度小于其温度区间的下界, 那么 qinth(int)就加上这个区间内该物流放出的热量
qintc(int)$[ord(int)=Tintint] = sum(j$[tcin(j)+hrat <= TINT(int+1) AND
tcout(j) + hrat >= TINT(int)], fc(j)*(TINT(int)-TINT(int+1)));
    在第 Tintint 个温度区间里, 遍历所有冷物流, 如果一个物流的出口温度大于等于其温度区间
    的上界, 进口温度小于其温度区间的下界, 那么 qintc(int)就加上这个区间内该物流吸收的热量
qint(int) = qinth(int) -qintc(int);
    该区间的净热量就是热物流放出的减去冷物流吸收的
);
Solve Minimumutility minimizing UMIN using lp;
    命令系统解决该 model, 使用线性规划方法, 得到 UMIN 的最小值
display TINT, UMIN.l , delta.l;
```

最后输出的结果是

```

PARAMETER TINT
int1 250.000, int2 240.000, int3 200.000, int4 190.000, int5 150.000
int6 80.000, int7 40.000, int8 30.000
---- 126 VARIABLE UMIN.L = 7.500
---- 126 VARIABLE delta.L
int1 9.000, int2 3.000, int3 4.000, int5 14.000, int6 12.000
int7 10.000, int8 10.000, int9 10.000, int10 10.000
```


运输模型 TRANSHIPMENT MODEL



这张图相当于只有一条热物流和一条冷物流传热，而系统里有 ij 个这样的图构成了一个网

现在我们来探索一个同时得到最小公用工程与换热网络的方法。

假设在问题表法中，我们分别只对一条热物流作上面的处理，而对冷物流我们只将其按照温度区间分段。此外我们还认为热量在每个区间内都由热物流传向冷物流，不存在低温热流区间向高温冷流区间传热。

那么，热物流和热公用的热平衡为：

$$\begin{cases} \delta_{i,0} = 0 & \delta_{U,0} \geq 0 \\ \delta_{i,j} = \delta_{i,j-1} + r_{i,j} - \sum_k s_{i,k,j} & \forall j = 1, \dots, n \end{cases}$$

冷物流的热平衡为

$$p_{k,j} = \sum_i s_{i,k,j} \quad \forall j = 1, \dots, n$$

其中 $r_{i,j}$ 和 $p_{k,j}$ 分别为热物流 i 和冷物流 k 在温度区间 j 中的热含量

现在问题就变成了：

限制条件： $\delta_{i,0} = 0 \quad i \neq U$

$$\delta_{i,j} = \delta_{i,j-1} + r_{i,j} - \sum_k s_{i,k,j} \quad \forall i \in U, \quad \forall j = 1, \dots, n$$

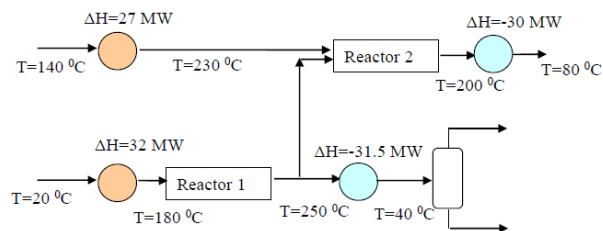
$$p_{k,j} = \sum_i s_{i,k,j} \quad \forall k, \forall j = 1, \dots, n$$

目标： $\delta_{U,0}$ 的最小值

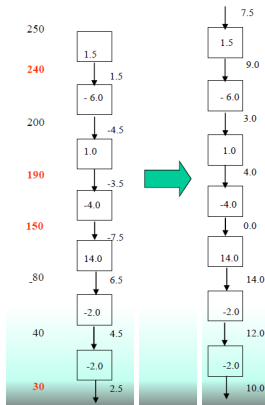
注意现在这个问题里，热物流包含了热公用，冷物流包含了冷公用，如果公用工程不止一条，就再加几条冷/热物流

我们先不考虑解决这个问题，我们先思考一下，这个问题解决了之后，是否可以给出一个换热网络？

来看下面这个实例（HRAT = 10°C）

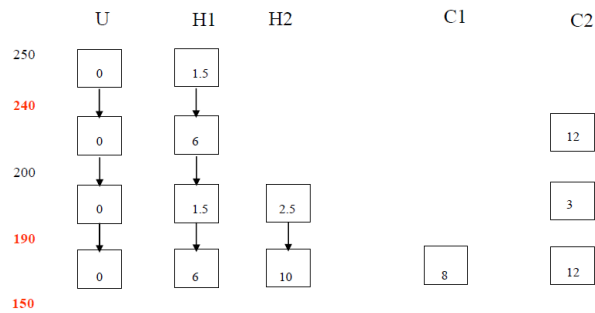


Stream	Type	Supply T (°C)	Target T (°C)	ΔH (MW)	F*cp (MW°C ⁻¹)
Reactor 1 feed	Cold	20	180	32.0	0.2
Reactor 1 product	Hot	250	40	-31.5	0.15
Reactor 2 feed	Cold	140	230	27.0	0.3
Reactor 2 product	Hot	200	80	-30.0	0.25



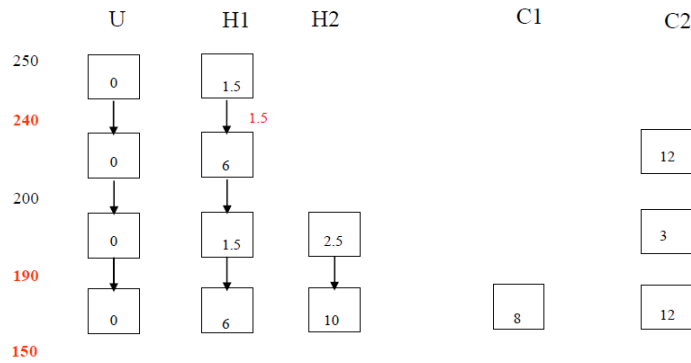
如果用问题表法，你会得到左边的级联图。

我们先只看夹点以上的部分，将总组合曲线拆回各自的曲线，然后得到五个串：



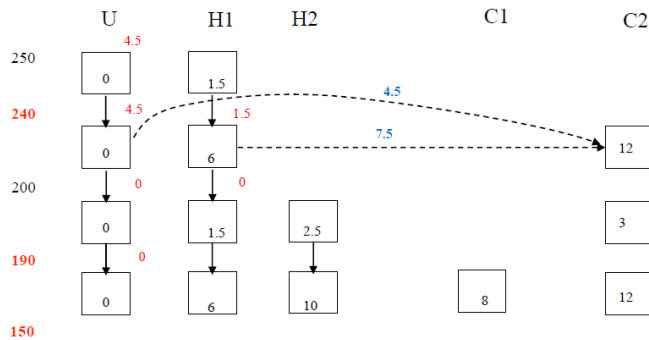
我们将尝试利用工程直觉找到解决方案，通过对比强调数学规划的重要性。

我们先从 H1 开始，第一个区间里没有冷物流，所以所有的热量都传下来

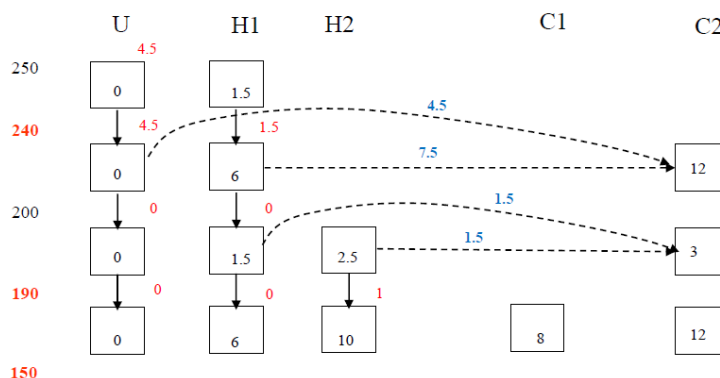


第二区间有冷物流了，将 H1 的 7.5 全部传给 C2，发现 C2 还需要 $12 - 7.5 = 4.5$ ，没有多余的热量了

这时候我们就要引入热公用，补足这 4.5。



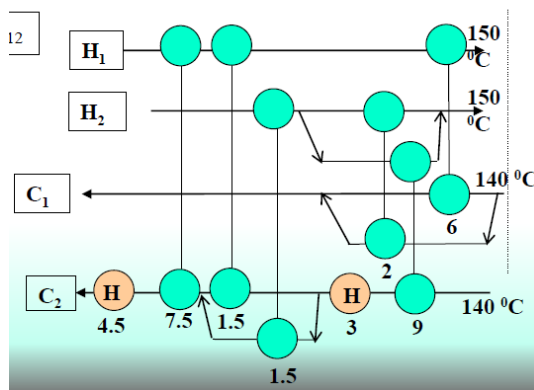
现在来到第三区间，有两个热物流，我们先用完 H1 的热量，如果不够，再用 H2 的



最后来到区间 4，同样先用 H1 满足 C1，不够再用 H2，然后是 C2，发现最后 C2 还差 3，还需要从热公用获得，那么 U 的起点就要+3，变成了 7.5，正好就是最小热公用量

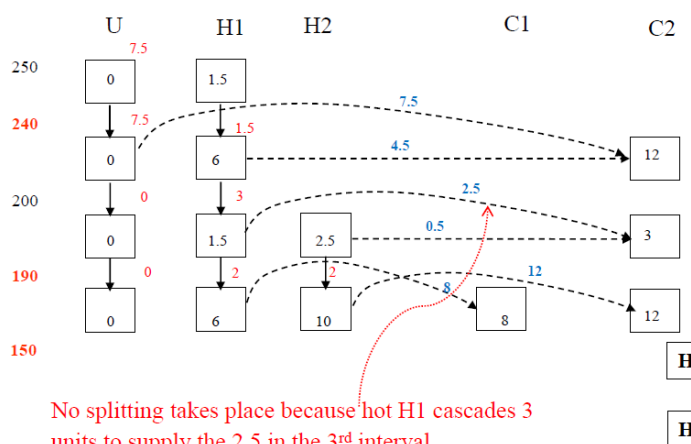
我们不仅得到了最小热公用，还得到了夹点之上的换热网络！

只是，它有那么亿点复杂……



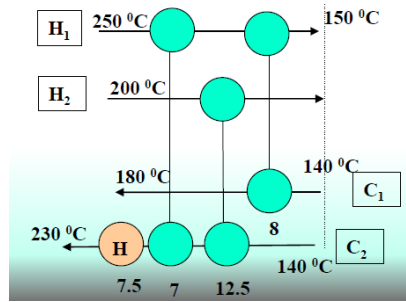
显然我们得到的分配表太复杂了，需要优化：

① C2 有 2 个 Heater，其中一个在相对较低的温度，显然我们可以将这个 Heater 移动到高位，和另一个 Heater 合并，这样 C2 就不用分流了。



② 不过 H2 和 C1 的分流就不能简单地通过检查移除了

我们回看一下通过 PDM 得到的换热网络，显然这个更加简单：



结论：我们需要让各个物流之间的连接数目达到最小。

COUNTING MATCHES

现在我们用新学到的数学规划来解决这个问题。

我们需要一个模型，这个模型可以给出单元数最小时的 $s_{i,k,j}$ 值

那我们怎么去数连接（match）的个数呢？我们需要一个布尔变量 $Y_{i,k}$ ，这意味着它只能取 0 或 1

然后通过让一个不等式成立，确定每个 Y 的值：

$$\sum_j s_{i,k,j} - \Gamma Y_{i,k} \leq 0$$

这意味着 $Y_{i,k} = 1$ 时，至少有一个温度区间内存在 i 与 k 的传热。

完整的模型为：

限制条件：① 对 $\forall i$ ， $\delta_{i,0} = 0$

$$\textcircled{2} \text{ 对 } \forall i, \forall j, \delta_{i,j} = \delta_{i,j-1} + r_{i,j} - \sum_k s_{i,k,j}$$

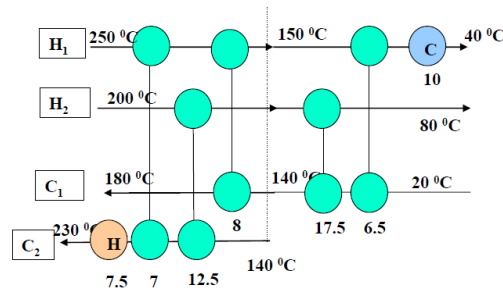
（ $r_{U,1}$ 是前面的模型求出的最小公用量）

$$\textcircled{3} \text{ 对 } \forall k, \forall j, p_{k,j} = \sum_i s_{i,k,j}$$

$$\textcircled{4} \text{ 对 } \forall i, \forall k, \sum_j s_{i,k,j} - \Gamma Y_{i,k} \leq 0$$

思考 这个模型必须按照夹点之上和夹点之下分开求解，为什么？

例子 如图



H1 和 C1 在夹点之上有一个匹配，在夹点之下也有一个匹配，如果不分开算，我们的算法会将这两个换热器记为一个 match。分开算的时候，结果是正确的。

结论 模型只能在夹点上下区域分开求解的时候给出正确的答案

附录：GAMS MODEL

SETS

```
I hot streams above pinch /U,H1,H2/
K cold streams above pinch /C1,C2/
J temperature intervals /J0*J3/;
```

SCALAR GAMMA /10000/;

TABLE R(I,J) load of hot stream I1 in interval K 温度区间j内热物流i放出的热

J1	J2	J3	J4	
U	7.5	0	0	0
H1	1.5	6	1.5	6
H2	0	0	2.5	10

TABLE P(K,J) load of cold stream K1 in interval J 每个温度区间每个冷物流需要的热

J0	J1	J2	J3	
C1	0	0	0	8
C2	0	12	3	12

VARIABLES

S(I,K,J) heat exchanged hot and cold streams
温度区间j里热物流i传给冷物流k的热

D(I,J) heat of hot streams flowing between intervals
温度区间j里热物流i传给下一个温度区间的热

Y(I,K) existence of match
热物流i与冷物流k的连接是否存在

Z total number of matches;

POSITIVE VARIABLE S

POSITIVE VARIABLE D

BINARY VARIABLE Y;

EQUATIONS

MINMATCH objective function - number of matches 连接数目 (目标函数)

HSBAL1(I,J) heat balances of hot stream I in INTERVAL J ne 1
第i热物流在第j个区间的热平衡 (j不为1) (限制条件)

HSBAL(I,J) heat balances of hot stream I in INTERVAL 1
第i热物流在第1个区间的热平衡 (限制条件)

CSBAL(K,J) heat balances of cold stream J in K
第k冷物流在第j个区间的热平衡 (限制条件)

HTINEQ1(I,K) heat transferred inequalities;
第i热物流与第k个冷物流的传热不等式 (限制条件)

MINMATCH .. Z =E= SUM((I,K), Y(I,K));

HSBAL1(I,J)\$(ORD(J) ne 1) .. D(I,J) - D(I,J 1) + SUM(K,S(I,K,J)) =E= R(I,J);

HSBAL(I,J)\$(ORD(J) eq 1) .. D(I,J) + SUM(K,S(I,K,J)) =E= R(I,J);

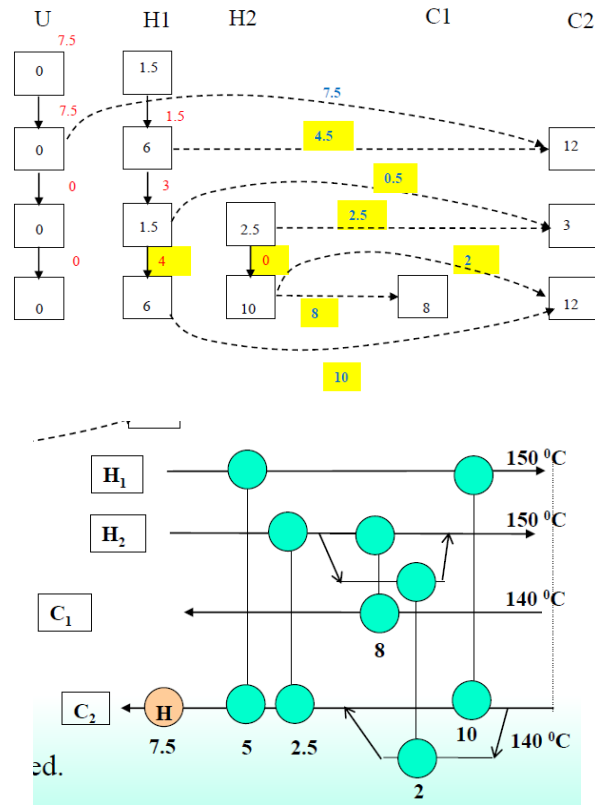
CSBAL(K,J).. SUM(I, S(I,K,J)) =E= P(K,J);

HTINEQ1(I,K) .. SUM(J, S(I,K,J)) - GAMMA*Y(I,K) =L= 0;

MODEL TSHIP /ALL/;

SOLVE TSHIP USING MIP MINIMIZING Z;

DISPLAY S.L, D.L, Y.L,Z.L;

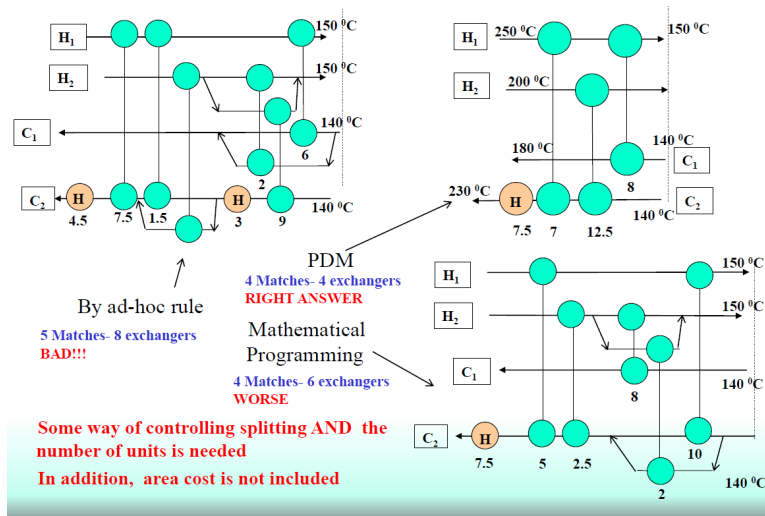


通过模型得到的结果和 PDM 有些不同，但它确实正确给出了 match 数。

这里 C2 产生分流是因为 H1 和 H2 都在最后一个区间有热量传给 C2。**H2 values cannot be moved to a sequential arrangement because the cascade is zero.**
(救命我看不懂这句话)

最后，H2 和 C2 在最后一个区间分流了，使用这个解决方案是无法改变的。

比较几个方法（包括上面没有提到的一些方法）：



我们需要一些控制分流和换热单元数的方法，此外，这里没有将面积费用包括进去

最后，我们回答一个问题：TRANSSHIPMENT MODEL 有用吗？

它确实给出了与 PDM 一样的物流匹配 (H1-C2, H2-C2, U-C2 和 H1-C1)

但是，它给出的换热单元数明显比 PDM 多。

所以，我们的结论是，TRANSSHIPMENT MODEL 可以帮助我们匹配物流，但不能给出最优的换热网络。

Part 3B

Superstructures & Synheat Model

上层建筑（？ 是这么翻译的吗）和热集成模型

Superstructure

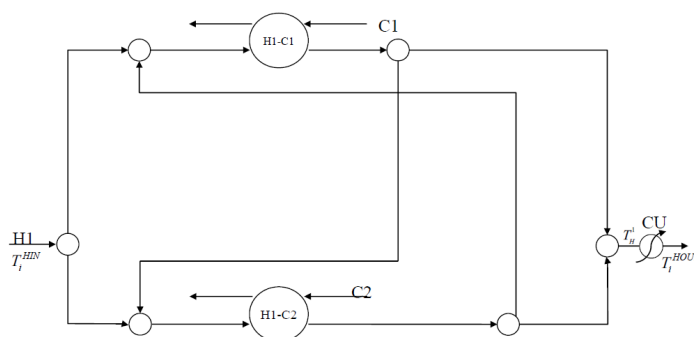
superstructure 是一种包括了选定的一组设备（equipment）和所有可能的连接（物流或者其它选定的 connection 类型）的流程图。在我们的问题中，设备是物流间换热器、加热器和冷却器，而连接随 superstructure 的具体模型而变

Superstructure 的模型有很多种，我们逐一介绍，然后对一种叫做 Stages Superstructure 的模型深入研究

1. Original Superstructure

这个模型里做出了一些假设：

- ① 公用工程加热/冷却都是在物流的尾端（箭头端）进行的
- ② 所有热物流和所有的冷物流只连接一次

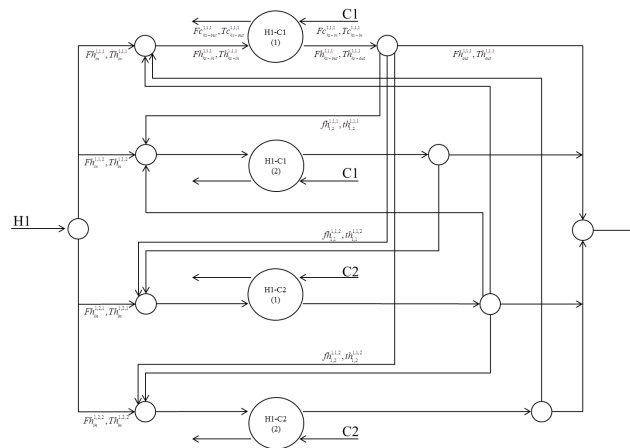


1H 2C Example

2. Generalized Superstructure

这个模型的假设是：

- ① 公用工程加热/冷却都是在物流的尾端（箭头端）进行的
- ② 所有热物流和所有的冷物流可以连接多次

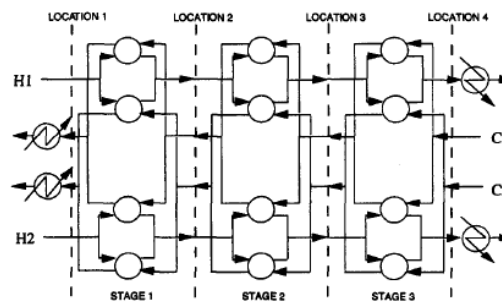


1H 2C Example

3. Stages Superstructure

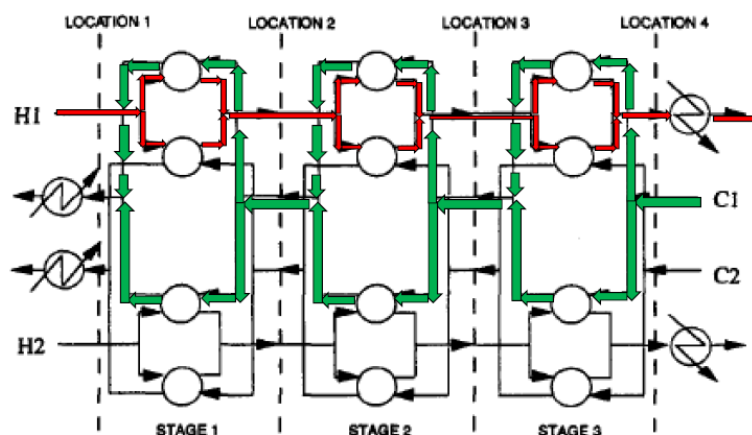
这个模型的假设是：

- ① 公用工程加热/冷却都是在物流的尾端（箭头端）进行的
- ② 阶段（stage）采用等温和非等温混合
- ③ 一对物流在一个 stage 只连接一次



Superstructure for 3 stages, two hot (H1,H2) and two cold (C1,C2) streams

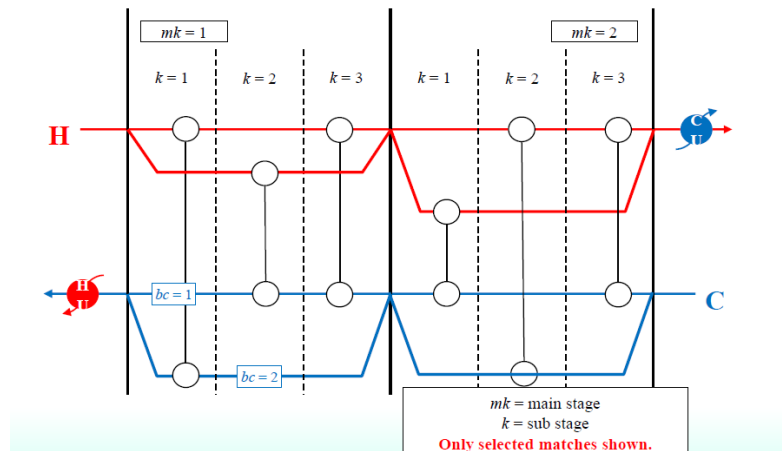
在每一个 stage 中，每条冷/热物流分流并于其它所有热/冷物流换热，下面标出了 H1 和 C1。



3. Stages Substages Superstructure

假设：

- ① 公用工程加热/冷却都是在物流的尾端（箭头端）进行的
- ② 阶段（stage）采用等温和非等温混合
- ③ 每个阶段和每个分支的每对流都有许多匹配



SUPERSTRUCTURE BASED MODELS 与 TRANSPORTATION MODELS 的比较

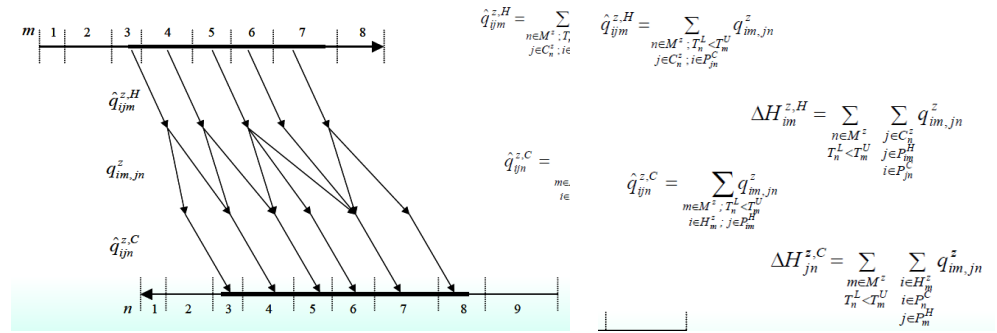
基于 superstructure 的模型是**非线性的**，这些模型

- Counts heat exchangers units and shells.
- Determines the area required for each exchanger unit or shell.
- Controls the total number of units.
- Determines the flow rates in splits.
- Handles non isothermal mixing.
- Identifies bypasses in split situations when convenient.
- Controls the temperature approximation (ΔT_{min}) when desired.
- Cannot address areas or temperature zones.
- Some allow multiple matches between two streams

而运输模型是**非线性的**，这些模型

- Counts heat exchangers units and shells.
- Determines the area required for each exchanger unit or shell.
- Controls the total number of units.
- Determines the flow rates in splits.
- Handles non isothermal mixing.
- Identifies bypasses in split situations when convenient.
- Controls the temperature approximation (ΔT_{min}) when desired.
- Can address areas or temperature zones.
- Allows multiple matches between two streams
- Was extended to retrofit and was used in industrial projects.
- Streams are divided in small temperature intervals

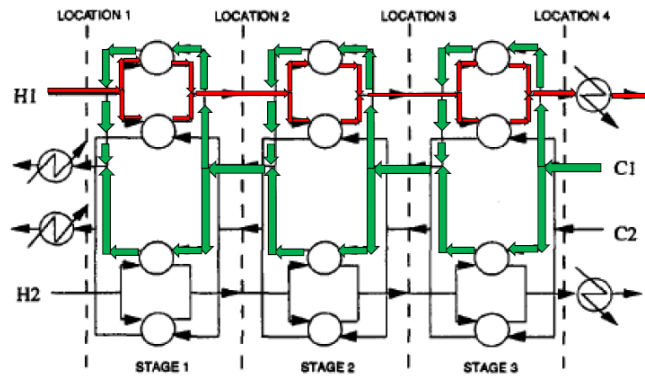
- Heat can be sent to **any** interval of lower temperature.



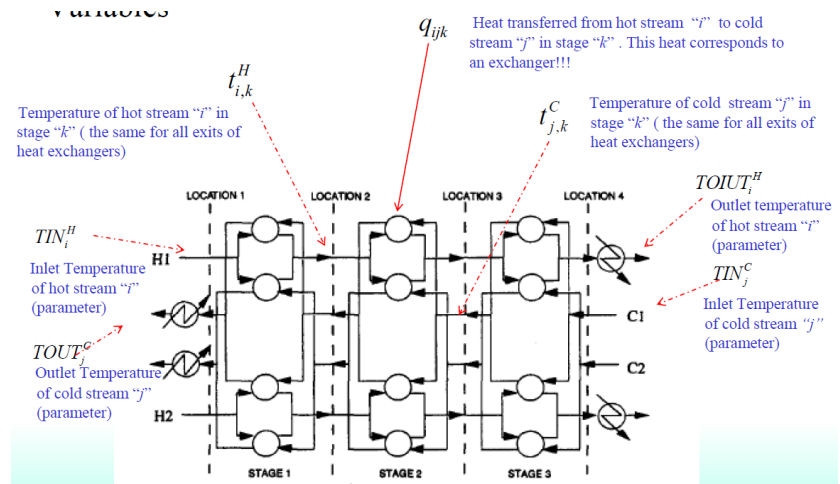
Stages Superstructure Model

我们重点关注 Stages Superstructure，建立一个数学规划模型。

一开始我们假设等温混合，这个假设稍后会被放宽。



变量



TIN_i^H — 热物流 i 的进口温度 $TOOUT_i^H$ — 热物流 i 的出口温度

$TOUT_j^C$ — 冷物流 j 的出口温度 TIN_j^C — 冷物流 j 的进口温度

$t_{i,k}^H$ — 热物流 i 进入阶段 k 时的温度

$t_{j,k}^C$ — 冷物流 j 离开阶段 k 时的温度

q_{ijk} — 热物流 i 和冷物流 j 在阶段 k 的换热量

约束条件

- ① 对每股物流，总的热平衡为

$$(TIN_i^H - TOUT_i^H)CP_i^H = \sum_k \sum_j q_{ijk} + qcu_i \quad qcu_i - \text{公用工程量}$$

$$(TOUT_i^C - TIN_i^C)CP_i^C = \sum_k \sum_j q_{ijk} + qhu_j \quad qhu_j - \text{公用工程量}$$

- ② 对每个阶段，热平衡为

$$(t_{ik}^H - t_{i(k+1)}^H)CP_i^H = \sum_j q_{ijk}$$

$$(t_{jk}^C - t_{j(k+1)}^C)CP_j^C = \sum_i q_{ijk}$$

- ③ 进入级 1 的热物流的温度就是进口温度 $TIN_i^H = t_{i1}^H$

进入级 NOK (最后一个级) 的冷物流的温度就是进口温度 $TIN_j^C = t_{j(NOK+1)}^C$

(冷物流进入 NOK 相当于离开 NOK+1 级)

- ④ 同一物流每级的温度都存在不等关系：

$$t_{ik}^H \geq t_{i(k+1)}^H \quad TOUT_i^H \geq t_{i(NOK+1)}^H$$

$$t_{jk}^C \geq t_{j(k+1)}^C \quad TOUT_j^C \geq t_{j,1}^C$$

- ⑤ 冷热公用满足

$$(t_{i(NOK+1)}^H - TOUT_i^H)CP_i^H = qcu_i$$

$$(TOUT_j^C - t_{j,1}^C)CP_j^C = qhu_j$$

- ⑥ 我们要对换热单元数目计数，所以设立三个布尔变量 z_{ijk} 、 zcu_i 、 zhu_j ，使得

$$q_{ijk} - \Omega z_{ijk} \leq 0 \quad qcu_i - \Omega zcu_i \leq 0 \quad qhu_j - \Omega zhu_j \leq 0$$

那么， $q > 0$ 时，对应的 z 为 1

- ⑦ 对于热物流 i 和冷物流 j 在级 k 的换热，传热温差应该使得

$$dt_{ijk} \leq t_{i,k}^H - t_{j,k}^C + \Gamma(1 - z_{ijk})$$

$$dt_{ij(k+1)} \leq t_{i,(k+1)}^H - t_{j,(k+1)}^C + \Gamma(1 - z_{ijk})$$

$$dtcu_i \leq t_{i(NOK+1)}^H - TOUT_{CU} + \Gamma(1 - zcu_i)$$

$$dthu_j \leq TOUT_{HU} - t_{j,1}^C + \Gamma(1 - zhu_j)$$

温差限制 $dt_{ijk} \geq EMAT$ 当 $z_{ijk} = 1$ (有换热) 的时候， $EMAT < dt_{ijk} \leq t_{i,k}^H - t_{j,k}^C$

这使得 $t_{i,k}^H - t_{j,k}^C > 0$

- ⑧ 目标函数 (公用费用 + 固定换热器设备费用)

$$O = \min \left\{ \begin{aligned} &CCU \sum_i qcu_i + CHU \sum_j qcu_j + \sum_i \sum_j \sum_k CF_{ij} z_{ijk} + \sum_i CF_{iCU} zcu_i + \sum_i CF_{jHU} zhu_j + \\ &\sum_i \sum_j \sum_k CV_{ij} A_{i,j,k} + \sum_i CV_i AC_i + \sum_j CV_j AH_j \end{aligned} \right\}$$

⑨ 换热面积

$$A_{i,j,k} = \frac{q_{i,j,k}}{\hat{U}_{i,j,k} LMTD_{i,j,k}} \quad AC_i = \frac{qcu_i}{\hat{U}_{c_i} LMTDCu_i} \quad AH_j = \frac{CV_j}{\hat{U}_{h_j} LMTDhu_j}$$

⑩ $LMTD_{ijk} = \frac{\Delta T_{i,j,k} - \Delta T_{i,j,k+1}}{\ln \frac{\Delta T_{i,j,k}}{\Delta T_{i,j,k+1}}}$ 如果采用 Chen 近似:

$$LMTD_{ijk} \approx \sqrt[3]{\Delta T_{i,j,k} \Delta T_{i,j,k+1} \frac{\Delta T_{i,j,k} + \Delta T_{i,j,k+1}}{2}}$$

除了换热面积相关的 Equations 以外，其余的限制都是线性的

目标函数可以被当成线性的

只要换热面积的 equation 在目标函数里被换成线性方程，那么限制就是线性的了

Sets

i hot streams /1 * 2/
j cold streams /1 * 2/;

Scalar nok number of stages in superstructure /2/;

Set

k temperature locations nok + 1 /1*3/ *k* 代表 location 的集合
st(k) stages; *st(k)*是 *k* 的子集，代表 stage (比 *k* 少 1 个)

Singleton Set

firstK(k) first temperature location; *k* 的单元素子集，代表第一个 location
lastK(k) last temperature location; *k* 的单元素子集，代表最后一个 location

st(k) = yes\$(ord (k) < card(k));

将顺序小于 *k* 集合元素个数的元素加入集合 *st*，也就是除了最后一个 location 都加进去

firstK(k) = yes\$(ord (k) = 1);

将顺序为 1 的元素加入集合 *st*，也就是第一个 location 都加进去

lastK(k) = yes\$(ord (k) = card(k));

将顺序为 *k* 集合元素个数 的元素加入集合 *lastK*，也就是最后一个 location

Parameters

fh(i) heat capacity flowrate of hot stream 热物流 *i* 的 CP
fc(j) heat capacity flowrate of cold stream 冷物流 *j* 的 CP
thin(i) supply temp. of hot stream 热物流 *i* 的进口温度
thout(i) target temp. of hot stream 热物流 *i* 的出口温度

tcin(j)	supply temp. of cold stream	热物流 j 的进口温度
tcout(j)	target temp. of cold stream	热物流 j 的出口温度
ech(i)	heat content hot i	热物流 j 的热含量
ecc(j)	heat content cold j	热物流 j 的热含量
hh(i)	stream individual film coefficient hot i	热物流 i 的膜系数
hc(j)	stream individual film coefficient cold j	冷物流 j 的膜系数
hucost	cost of heating utility	热公用的成本
cucost	cost of cooling utility	冷公用的成本
unitc	fixed charge for exchanger	换热器固定费用
acoeff	area cost coefficient for exchangers	换热器单位面积费用
hucoeff	area cost coefficient for heaters	热公用单位面积费用
cucoeff	area cost coefficient for coolers	冷公用单位面积费用
aexp	cost exponent for exchangers	换热器费用指数
hhu	stream - individual film coefficient hot utility	换热器费用指数
hcu	stream - individual film coefficient cold utility	换热器费用指数
thuin	inlet temperature hot utility	热公用进口温度
thuout	outlet temperature hot utility	热公用出口温度
tcuin	inlet temperature cold utility	冷公用进口温度
tcuout	outlet temperature cold utility	冷公用出口温度
gamma(i,j)	upper bound of driving force	推动力上限
a(i,j,k)	area for exchanger for match ij in interval k (chen approx.)	用 <i>chen</i> 近似法得到的换热面积 ijk
al(i,j,k)	area calculated with log mean	用 <i>log</i> 法计算的换热面积
acu(i)	area coolers	冷公用换热面积
ahu(j)	area heaters	热公用换热面积
tmapp	minimum approach temperature	最小??? 温度
costheat	cost of heating	热公用成本
costcool	cost of cooling	冷公用成本
invcost	investment cost;	总投资成本

Binary Variables

$z(i,j,k)$	热物流 i 和冷物流 j 在 k 级处是否有换热
$zcu(i)$	热物流 i 在 k 级处是否有冷公用
$zhu(j);$	冷物流 j 在 k 级处是否有热公用

Positive Variables

$th(i,k)$	temperature of hot stream i as it enters stage k	热物流 i 进入 k 级时的温度
$tc(j,k)$	temperature of cold stream j as it leaves stage k	冷物流 j 进入 k 级时的温度
$q(i,j,k)$	energy exchanged between i and j in stage k	热物流 i 和冷物流 j 在 k 级时的换热量

qc(i) energy exchanged between i and the cold utility

热物流 i 与冷公用的换热量

qh(j) energy exchanged between j and the hot utility

冷物流 j 与热公用的换热量

dt(i,j,k) approach between i and j at location k

热物流 i 与冷物流 j 在节点 k 的温差

dtcu(i) approach between i and the cold utility

热物流 i 与冷公用的温差

dthu(j) approach between j and the hot utility;

冷物流 i 与热公用的温差

Variable cost hen and utility cost;

Equations

teh(i) total energy exchanged by hot stream i

计算热物流 i 全程传递的所有热

tec(j) total energy exchanged by cold stream j

计算向冷物流 j 全程传递的所有热

ec(j,k) energy exchanged by cold stream j in stage k

计算冷物流 j 在级 k 传递的热

eh(i,k) energy exchanged by hot stream i in stage k

计算热物流 i 在级 k 传递的热

eqc(i) energy exchanged by hot stream i with the cold utility

计算热物流 i 向冷公用传递的热

eqh(j) energy exchanged by cold stream j with the hot utility

计算热公用向冷物流 j 传递的热

month(i,k) monotonicity of th

montc(j,k) monotonicity of tc

monthl(i) monotonicity of th k = last

montcf(j) monotonicity of tc for k = 1

tinh(i) supply temperature of hot streams

tinc(j) supply temperature of cold streams

logq(i,j,k) logical constraints on q

logqh(j) logical constraints on qh(j)

logqc(i) logical constraints on qc(i)

logdth(i,j,k) logical constraints on dt at the hot end

logdtc(i,j,k) logical constraints on dt at the cold end

logdtcu(i) logical constraints on dtcu

logdthu(j) logical constraints on dthu

obj 'objective function';

teh(i).. (thin(i) - thout(i))*fh(i) =e= sum((j,st), q(i,j,st)) + qc(i);

热物流 i 在全过程放出的热 等于 热物流 i 在所有 stage 与所有冷物流 j 交换的热之和

tec(j).. (tcout(j) - tcin(j))*fc(j) =e= sum((i,st), q(i,j,st)) + qh(j);

冷物流 j 在全过程吸收的热 等于 冷物流 j 在所有 stage 与所有热物流 i 交换的热之和

$$eh(i,k)\$st(k).. fh(i)*(th(i,k) - th(i,k+1)) =e= sum(j,q(i,j,k));$$
 当 k 属于 st 集合时 (即不是最后一个 location) 时
 热物流 i 在 stage k 放出的热 等于 热物流 i 在 stage k 与所有冷物流 j 交换的热之和

$$ec(j,k)\$st(k).. fc(j)*(tc(j,k) - tc(j,k+1)) =e= sum(i,q(i,j,k));$$
 当 k 属于 st 集合时 (即不是最后一个 location) 时
 冷物流 j 在 stage k 吸收的热 等于 冷物流 j 在 stage k 与所有热物流 i 交换的热之和

$$eqc(i).. fh(i)*(th(i,lastK) - thout(i)) =e= qc(i);$$
 当 k 是最后一个 location 时
 进入 lastk 的热物流 i 与出口温相比的能量变化 等于 热物流 i 与冷公用工程的交换量

$$eqh(j).. fc(j)*(tcout(j) - tc(j,firstK)) =e= qh(j);$$
 当 k 是第一个 location 时
 离开 firstk 的冷物流 j 与出口温相比的能量变化 等于 冷物流 j 与热公用工程的交换量

$$tinh(i).. thin(i) =e= th(i,firstK);$$
 热物流 i 的进口温度 等于 热物流 i 在第一个 location 时的温度
 (即进入第一个 stage 时的温度)

$$tinc(j).. tcin(j) =e= tc(j,lastK);$$
 冷物流 j 的进口温度 等于 冷物流 j 在最后一个 location 时的温度

$$month(i,k)\$st(k).. th(i,k) =g= th(i,k+1);$$
 热物流 i 进入每一 stage 时的温度都不低于下一个 stage 时的温度
 (包括最后一个 stage, 此时 $k+1$ 为最后一个 location)

$$montc(j,k)\$st(k).. tc(j,k) =g= tc(j,k+1);$$
 冷物流 j 离开每一 stage 时的温度都不低于离开上一个 stage 时的温度
 (冷物流是从大 stage 往小 stage 流的)
 (包括最后一个 stage, 此时 $k+1$ 为最后一个 location)

$$monthl(i).. th(i,lastK) =g= thout(i);$$
 热物流 i 进入最后一个 location 时的温度不低于最终的出口温度

$$montcf(j).. tcout(j) =g= tc(j,firstK);$$
 冷物流 j 离开第一个 location 时的温度不低于最终的出口温度

$$logq(i,j,k)\$st(k).. q(i,j,k) - \min(ech(i), ecc(j))*z(i,j,k) =l= 0;$$
 热物流 i 与冷物流 j 在 stage k 的换热量大于 0 时, $q(i,j,k)$ 的值为 1, 否则为 0

$$logqc(i).. qc(i) - ech(i)*zcu(i) =l= 0;$$
 热物流 i 与冷公用工程交换量 > 0 时, $zcu(i)$ 的值为 1, 否则为 0

$$logqh(j).. qh(j) - ecc(j)*zhu(j) =l= 0;$$
 冷物流 j 与热公用工程交换量 > 0 时, $zhu(j)$ 的值为 1, 否则为 0

$$logdth(i,j,k)\$st(k)..$$

$$dt(i,j,k) =l= th(i,k) - tc(j,k) + \gamma(i,j)*(1 - z(i,j,k));$$
 热物流 i 与冷物流 j 在 k 阶有热交换时, 限制热物流进口端传热温差

$$logdtc(i,j,k)\$st(k)..$$

$$dt(i,j,k+1) =l= th(i,k+1) - tc(j,k+1) + \gamma(i,j)*(1 - z(i,j,k));$$
 热物流 i 与冷物流 j 在 k 阶有热交换时, 限制冷物流进口端传热温差

$$logdthu(j).. dthu(j) =l= (thout - tc(j,firstK));$$
 冷物流 j 有用热公用加热时, 限制冷物流进口端传热温差

$$logdtcu(i).. dtcu(i) =l= th(i,lastK) - tcuout;$$

冷物流 j 有用热公用加热时，限制冷物流进口端传热温差

```
obj.. cost =e= unitc*(sum((i,j,st),z(i,j,st)) + sum(i,zcu(i)) + sum(j,zhu(j)))
      + acoeff*sum((i,j,k),(q(i,j,k)*((1/hh(i))
      + (1/hc(j)))) / (((dt(i,j,k)*dt(i,j,k+1)*(dt(i,j,k) +
dt(i,j,k+1))/2
      + 1e-6)**0.33333) + 1e 6) + 1e 6)**aexp)
+ hucoeff*(sum(j,(qh(j)*((1/hc(j)) + 1/hhu)) / (((thu in -
tcout(j))*dthu(j)*((thu in tcout(j) + dthu(j))/2)
      + 1e-6)**0.33333) + 1e 6)**aexp)
+ cucoeff*sum(i,(qc(i)*((1/hh(i)) + (1/hcu)) /
      (((thout( tcuin)*dtcu(i)*(thout(i) tcuin + dtcu(i))/2
      + 1e-6)**0.33333) + 1e 6)**aexp)
      + sum(j,qh(j)*hucost) + sum(i,qc(i)*cucost);
```

计算总成本

下面输入基础数据

Stream	T_{IN} (K)	T_{OUT} (K)	h (kW/m ² K)	F_{cp} (kW/K)
H1	650.0	370.0	1.0	10.0
H2	590.0	370.0	1.0	20.0
C1	410.0	650.0	1.0	15.0
C2	350.0	500.0	1.0	13.0
CU	300.0	320.0	1.0	-
HU	680.0	680.0	5.0	-
HE cost	5500 + 150 Area (\$/y)			
Utility cost	80 q _{hu} + 15 q _{cu} (\$/kW _y)			
EMAT _{Min}	10 K			

```
thin('1') = 650; thout('1') = 370; fh('1') = 10; hh('1') = 1;
thin('2') = 590; thout('2') = 370; fh('2') = 20; hh('2') = 1;
```

```
tcin('1') = 410; tcout('1') = 650; fc('1') = 15; hc('1') = 1;
tcin('2') = 350; tcout('2') = 500; fc('2') = 13; hc('2') = 1;
```

```
hucost = 80; hucoeff = 150; thu in = 680; thuout = 680; hhu = 5;
cucost = 15; cucoeff = 150; tcuin = 300; tcuout = 320; hcu = 1;
unitc = 5500; acoeff = 150;
```

```
aexp = 1; tmapp = 10;
```

规定自变量边界

```
dt.lo(i,j,k) = tmapp; 传热温差不低于 tmapp
```

```
dthu.lo(j) = tmapp;
```

```
dtcu.lo(i) = tmapp;
```

```
th.up(i,k) = thin(i);
```

```
th.lo(i,k) = thout(i);
```

```
tc.up(j,k) = tcout(j);
```

```
tc.lo(j,k) = tcin(j);
```

初始化

```
th.l(i,k) = thin(i);
```

```
tc.l(j,k) = tcin(j);
```

```

dthu.l(j) = thuout - tcin(j);
dtku.l(i) = thin(i) tcuout;
ech(i) = fh(i)*(thin(i) - thout(i));
ecc(j) = fc(j)*(tcout(j) tcin(j));
gamma(i,j) = max(0,tcin(j)-thin(i), tcin(j)-thout(i),tcout(j)-thin(i), tcout(j)
-thout(i));
dt.l(i,j,k) = thin(i)-tcin(j);
q.l(i,j,k)$st(k) = min(ech(i),ecc(j));
Model super /all/;
option optCr = 0, limRow = 0, limCol = 0, solPrint = off, sysOut = off;
Solve super using minlp minimizing cost;
计算并输出一些重要的数据
传热面积计算 (用 Chen 近似)
a(i,j,k)$st(k) = q.l(i,j,k) * ((1/hh(i)) + (1/hc(j))) /
(2/3*sqrt(dt.l(i,j,k)*dt.l(i,j,k+1)) + 1/6*(1e-8 + dt.l(i,j,k) +
dt.l(i,j,k+1)));
传热面积计算 (用对数法)
al(i,j,k)$st(k) = (q.l(i,j,k)*((1/hh(i)) + (1/hc(j)))) /
(dt.l(i,j,k)*dt.l(i,j,k+1) * (dt.l(i,j,k) + dt.l(i,j,k+1))/2)**0.33333;
display a, al;
公用工程传热面积计算 (用对数法)
ahu(j) = (qh.l(j)*((1/hc(j)) + (1/hhu)))/(((thu in - tcout(j))*dthu.l(j) *
((thu in - tcout(j)+dthu.l(j))/2) + 1e 6)**0.33333);
acu(i) = (qc.l(i)*((1/hh(i)) + (1/hcu)))/(((thout(i) - tcuin)*dtku.l(i) *
(thout(i) -tcuin + dtku.l(i))/2 + 1e 6)**0.33333));
display acu, ahu;
公用工程费用
costheat = sum(j,qh.l(j)*hucost);
costcool = sum(i,qc.l(i)*cucost);
display costheat, costcool;
invcost = cost.l - costheat - costcool;
display invcost;

```

输出结果

```

---- 207 PARAMETER a area for exchanger for match ij in interval k (chen
approx.)

```

	1	2
1.1	71.120	
1.2		69.257
2.1		140.182

```

---- 207 PARAMETER al area calculated with log mean

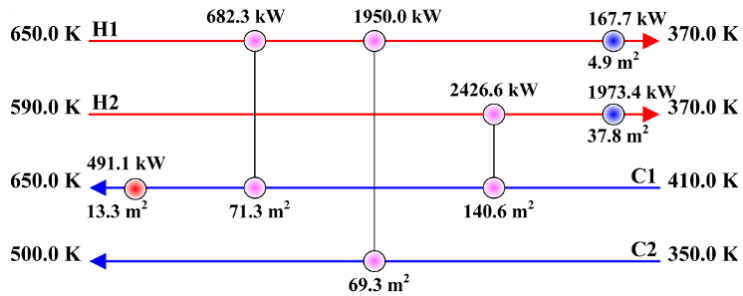
```

	1	2
1.1	71.345	
1.2		69.306

```

2.1      140.602
---- 216 PARAMETER acu area coolers
1 4.906, 2 37.808
---- 216 PARAMETER ahu area heaters
1 13.287
---- 222 PARAMETER costheat = 39291.945 cost of heating
      PARAMETER costcool = 32117.240 cost of cooling
---- 227 PARAMETER invcost = 83588.150 investment cost

```



SubOptimal Solutions

为了检验下程序是不是给出了最佳设计，我们可以让它给出第二个答案。
我们如何得到第二种设计（尽管它不是最优的）？

答：令 Z_l 为包含当前答案中等于 1 的 $z(i, j, k)$ ， $zcu(i)$ 和 $zhu(j)$ 的集合。

i,j	k='1'	k='2'
	$z(i,j,'1')$	$z(i,j,'2')$
1,1	1	2
1,2		1
2,1		1

i	zcu
1	1
2	1

j	zhu
1	1
2	

这样我们就可以加入一个“排除”用的限制条件：

$$\left(\sum_{i,j,k} z_{i,j,k} + \sum_i zcu_i + \sum_j zhu_j \right) (i,j,k \in Z_l) - \left(\sum_{i,j,k} z_{i,j,k} + \sum_i zcu_i + \sum_j zhu_j \right) (i,j,k \notin Z_l) \leq \text{Card}(Z_l) - 1$$

代码为

```

Equation exclusion_1;
exclusion_1.. z('1','1','1') + z('1','2','2') + z('2','1','2') +
zcu('1') + zcu('2') + zhu('1') -z('1','1','2')-z('1','2','1')-
z('2','1','1')-z('2','2','1')-z('2','2','2')-zhu('2')) =1= 5;

```

那么，对于我们得到的一个答案：

$$(z_{111} + z_{122} + z_{212} + zcu_1 + zcu_2 + zhu_1) - (z_{112} + z_{121} + z_{211} + z_{221} + z_{222} + zhu_2) \leq 6 - 1 = 5$$

那么程序之前写好的代码时，肯定会尝试第一个答案，这时候因为

$$(1+1+1+1+1+1) - (0+0+0+0+0+0) = 6 > 5$$

不满足新加上的条件，这样程序就不会得到重复的答案了

结果为

```

---- 211 PARAMETER a area for exchanger for match ij in interval k (chen
approx.)
      1      2
1.1 71.120
1.2      69.257
2.1      140.182
---- 220 PARAMETER acu area coolers
1 4.906, 2 37.808
---- 220 PARAMETER ahu area heaters
1 13.287
---- 214 VARIABLE zhu.L
1 1.000, 2 1.000
---- 226 PARAMETER costheat = 39291.945 cost of heating
      PARAMETER costcool = 32117.240 cost of cooling
---- 231 PARAMETER invcost = 89088.150 (83588.150) investment cost

```

结果是，多了一个换热单元，但是传热量为 0，显然这里出了点小问题

我们需要再加一些限制来防止 $z=1$ ， $q=0$ 这种情况发生：

$$q_{i,j,k} \geq \varepsilon z_{i,j,k} \quad qcu_i \geq \varepsilon zcu_i \quad qhu_j \geq \varepsilon zhu_j$$

ε 是一个很小的数字（一般定为最小传热量）这样当 $z=1$ 时 q 就必须不为 0 了

```

parameter epsilon;
epsilon = 10;
Equation minq, minqcu, minqhu;
minq(i,j,k) .. q(i,j,k) - epsilon * z(i,j,k) =g= 0;
minqcu(i) .. qc(i) - epsilon * zcu(i) =g= 0;
minqhu(j) .. qh(j) - epsilon * zhu(j) =g= 0;

```

最后得到的结果为：

```

---- 222 VARIABLE z.L      与第一个答案相同
      1      2
1.1 1.000
1.2      1.000
2.1 1.000
---- 222 VARIABLE zcu.L    少了一个冷却器
2      1.000
---- 222 VARIABLE zhu.L    与第一个答案相同
1 1.000
---- 222 PARAMETER a area
      1      2
1.1 80.538
1.2      102.084
2.1      85.294

```

```

---- 231 PARAMETER acu area coolers
2 40.722
---- 231 PARAMETER ahu area heaters
1 14.827                                q = 575 之前的结果是 q = 491
---- 237 PARAMETER costheat = 46000.000 cost of heating
      PARAMETER costcool = 33375.000 cost of cooling
---- 242 PARAMETER invcost = 76134.573 investment cost
VARIABLE cost.L = 155509.573 hen and utility cost
之前的结果是 q = 154997.335

```

这说明，在给出了限制之后，程序就给不出最优结果了

所以，最后的最优结果是

```

      1      2
1.1 71.120
1.2      69.257
2.1      140.182
---- 216 PARAMETER acu
1 4.906, 2 37.808
---- 216 PARAMETER ahu
1 13.287

```

Part 3C

HEN Synthesis Using Enumeration & Golden search

用枚举法和黄金搜索合成换热网络

枚举法 Enumeration

枚举法是计算机编程中一类重要的思想方法。在我们的课程中，它的目标是得到所有可能的结构。这意味着与传热相关的所有可能且可行的匹配组合 $z(i,j,k)$ 都不等于零。

在枚举法中，我们就不在意最优的方案了。我们只想要得到不同且合理的最小数目连接排列。

① 运行 Synheat，但它不包含计算面积的 equation 以及不同的目标函数。

然后确定匹配的数目

② 应用 3B 提到的“排除”约束

③ 注意错误的备选项

$$\text{新的目标函数为 match 数目 } O = \min \left\{ \sum_i \sum_j \sum_k z_{ijk} + \sum_i zcu_i + \sum_j zhu_j \right\}$$

$$\text{限制 match 数量的约束为 } \sum_i \sum_j \sum_k z_{ijk} + \sum_i zcu_i + \sum_j zhu_j \leq N_{\min}$$

现在结果模型就是线性的了（可以使用“mip”求解器），因为不需要面积了。

对于 $z_{i,j,k}$ 都已经确定的结构，如果热公用量固定，即

$$\sum_j qhu_j = E$$

则当 $N = N_{\min}$ 时，得到的结果 q_{ijk} 将会是唯一值（记住 $z_{i,j,k}$ 是固定的），并且

$$q_{ijk} - \Omega z_{ijk} \leq 0 \quad qcu_i - \Omega zcu_i \leq 0 \quad qhu_j - \Omega zhu_j \leq 0$$

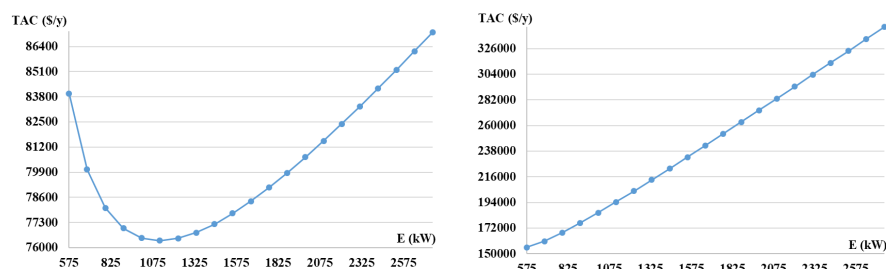
一旦得到了 q_{ijk} ，对于一个确定的结构和不同的 E，就可以计算面积

$$A_{i,j,k} = \frac{q_{i,j,k}}{\hat{U}_{i,j,k} LMTD_{i,j,k}} \quad AC_i = \frac{qcu_i}{\hat{U}_{c,i} LMTDC_{u_i}} \quad AH_j = \frac{qhu_j}{\hat{U}_{h,j} LMTDH_{u_j}}$$

然后就能得到 TAC (Total annualized Cost)

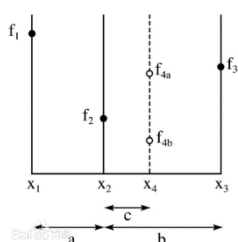
$$TAC = \left\{ CCU \sum_{i \in HP} qcu_i + CHU \sum_{j \in CP} qhu_j + \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} CF_{ij} z_{ijk} + \sum_{i \in HP} CF_{iCU} zcu_i + \sum_{j \in CP} CF_{jHU} zhu_j + \right. \\ \left. ca \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} A_{i,j,k}^a + ca \sum_{i \in HP} AC_i^{ac} + ca \sum_{j \in CP} AH_j^{ah} \right\}$$

通过大量实例，我们猜想: TAC 有且仅有一个极小值，且没有极大值



那么，如何用最少的时间找到这个值，从而确定最佳 E 呢？

黄金分割搜索算法 Golden Search Algorithm



黄金分割搜索是一种算法，用于找单峰函数（就是只有一个极值点的函数）的唯一最值。它类似于二分法，但二分法缩小查找区间使用的是平均值，而黄金分割搜索使用的黄金分割比。

黄金分割比是指一条线段 AB 被点 C 分为了 AC、BC 两个部分，其中 $AC < BC$ ，使得

$$\frac{AC}{BC} = \frac{BC}{AB}$$

简单解一个二次方程，就得到这个比例为 $\varphi = \frac{\sqrt{5}-1}{2}$ ，这个数字很神奇，它加

上 1 就是它的倒数 $\frac{\sqrt{5}+1}{2}$ ，也就是 $\frac{BC}{AC} = \frac{AB}{BC}$

如果 $AC > BC$ ，可以得到另一个 C 点，也就是说，**一条线段上存在 2 个黄金分割点。**

黄金分割算法的具体的步骤如下：

1. 确定目标函数 $f(x)$ ，限定搜索区间为 x_1 到 x_2 ，得到函数值 $f(x_1)$ 和 $f(x_2)$

在这里， $f(x)$ 就是 TAC， x 为 E。但是我们没有直接从 E 得到 TAC 的解析表达式，需要将 E 送进 Synheat 里得到 q_{ijk} ，然后计算 TAC。

2. 找到区间内的黄金分割点 x_3 ，并计算 $f(x_3)$

x_3 应当将原有的区间分为两段 $x_3 - x_1$ 和 $x_2 - x_3$ ，这两段的长度比值为黄金比例 φ 或 $1/\varphi$

3. 确定三元组 $f(x_1)$ ， $f(x_2)$ ， $f(x_3)$ 是否满足收敛准则（这里涉及数值分析了）

如果是，则从这个三元组中得到最小值及对应的 X 作为结果，算法结束
否则继续进行下一步

4. 在区间里找到另一个黄金分割点 x_4 ，并计算函数值 $f(x_4)$

那么 x_1, x_4, x_2 满足黄金比例： $\frac{x_2 - x_4}{x_4 - x_1} = \varphi$

并且 $\frac{x_4 - x_3}{x_2 - x_4} = \frac{x_4 - x_3}{x_3 - x_1} = \varphi$ 都是黄金比例，非常神奇

5. 舍去四个 $f(x)$ 中的最大值对应的 x ，剩下的三个 x 将是下一次迭代的三元组

最小值一定是 $f(x_3)$ 和 $f(x_4)$ 中的一个，如果 $f(x_3)$ 最小，那么最小值就会在

x_1, x_3, x_4 区间内, 如果 $f(x_4)$ 最小, 那么最小值就会在 x_3, x_4, x_2 区间内, 不管是哪个区间, 它们都是黄金比例, 然后将这三个点替代原来的三元组。

6. 回到步骤 3

这时三个点代替原来的 x_1, x_2, x_3 , 但是要注意, $\frac{x_3 - x_1}{x_2 - x_3}$ 可能是 φ , 可能是 $1/\varphi$, 这是由上一次迭代的结果确定了, 所以设计算法时, 应当设置条件语句判断要找哪一个黄金分割点

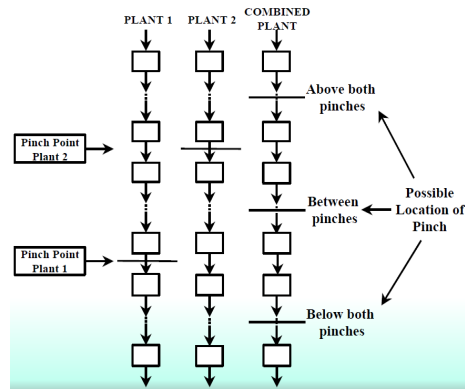
Part 3D

Interplant Heat Integration

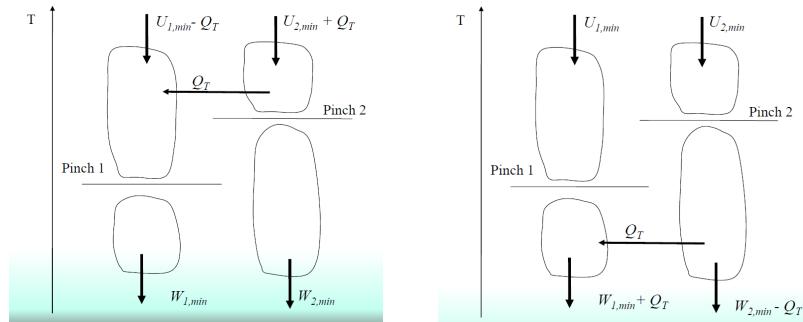
厂际热集成

TOTAL SITE INTEGRATION

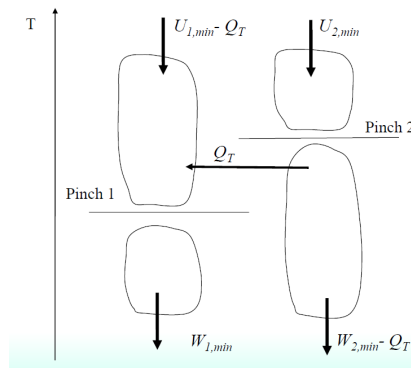
考虑两套装置, 我们想知道什么条件下一套装置可以传热给另一套来减小能耗。两套装置都有一个夹点, 将整个级联图分成了 3 个区域。我们将两个夹点包围的区域称为 inter pinch region



简单分析得出, 在 inter pinch region 的上面和下面热集成无法实现任何有效节能

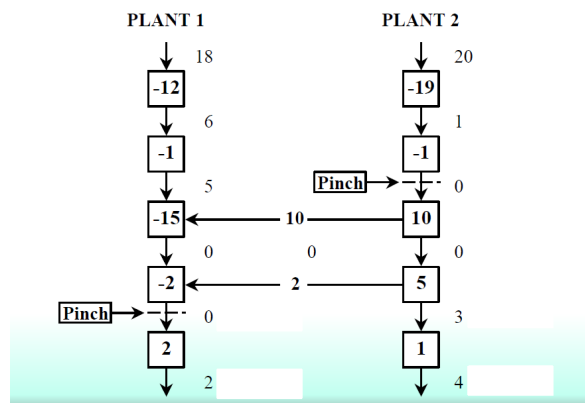


只有在两个夹点之间集成, 才能有效节能。此时装置 2 相当于热源, 装置 1 相当于热阱。

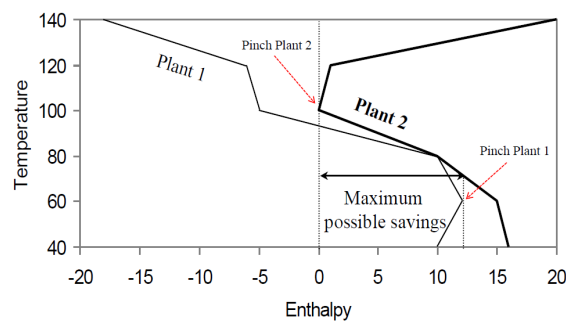


GRAND COMPOSITE CURVES

考虑下面这个例子，我们有没有直观地看出节能目标的途径？



答案是 GCC 图。GCC 图可以直观地给出一套装置在某个温度下的净余热量和需求热量。我们将两套装置的 GCC 组合起来，就可以得到总图象。组合方法是将装置 1 的 GCC 转 180°（或者镜像对称），然后让两条 GCC 尽可能重叠，但不能有交叉。

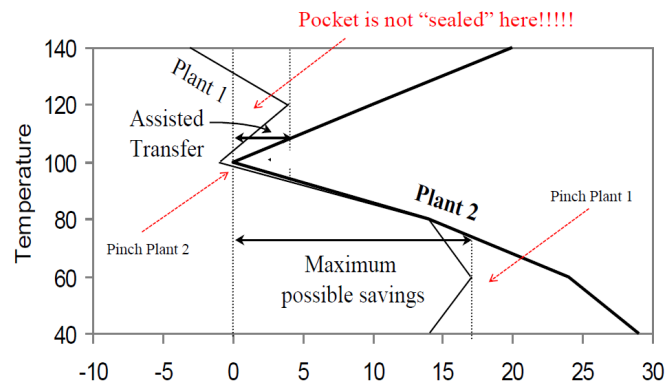
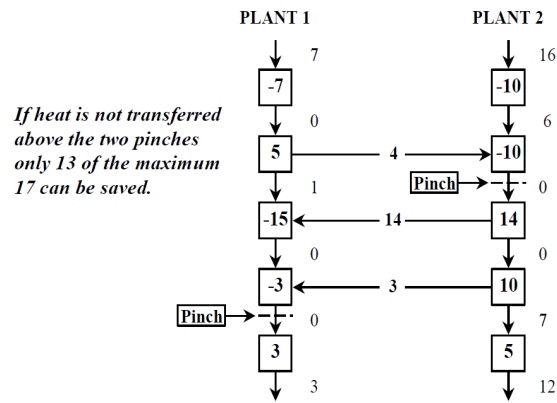


通过组合的 GCC 图就可以看出，厂际热集成最多可以节约 12 单位能量

ASSISTED HEAT TRANSFER

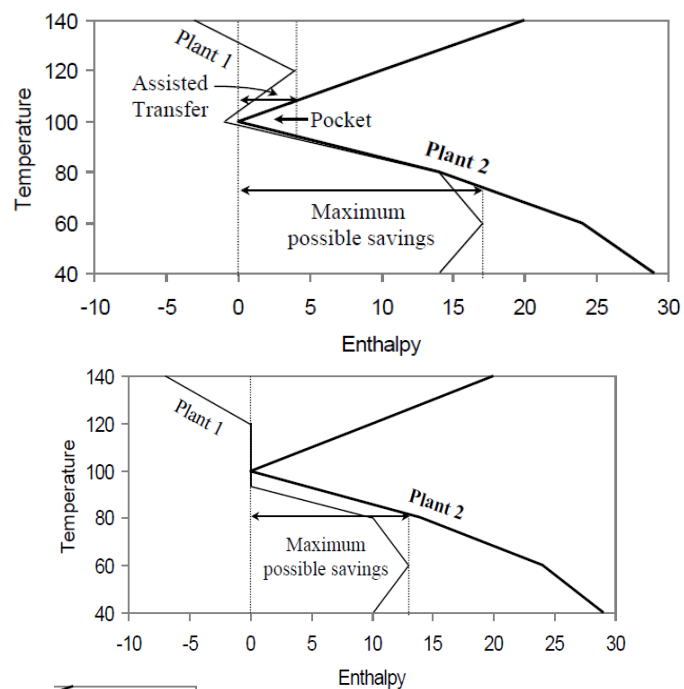
再看下面这个例子：

通过级联图可以看出，最多可以节约 13 能量，但是从 GCC 图上看，最多可以节约 17 能量，怎么回事？



这次是 GCC 出问题了，可以看到装置 1 有一个 Pocket 进入了重叠区间，但是，pocket 的意义是同一装置某一级多出来的热量用于下面的需要额外热量的级，这部分热量是自给自足的部分，不能给其它装置。所以在组合之前，我们要先将 GCC 所有的 pocket “密封”，也就是变成一条竖线，再做处理。

注意，装置 1 的 GCC 图是翻转过的，向左凸出来的才是 pocket。



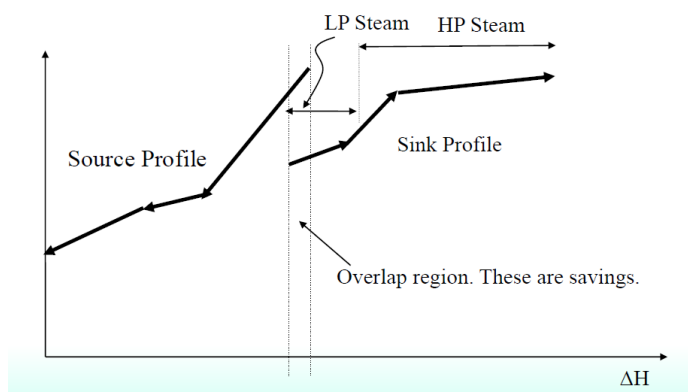
修正之后就与级联图对上了。

USING UTILITIES AS INTERMEDIARIES

考虑到厂际热集成的实现方式，可能都会想到将工艺物流运输到另一套装置去，与它的对象“见面”。但是，装置的距离往往很远，这么做有的时候会很不划算。这时就会考虑引入中间介质，它像快递员一样，将热量从一套装置带到另一套装置。中间介质的性质一般都很安全，运输成本低，但是传热也会出现损耗，所以热回收效率不如直接传热。

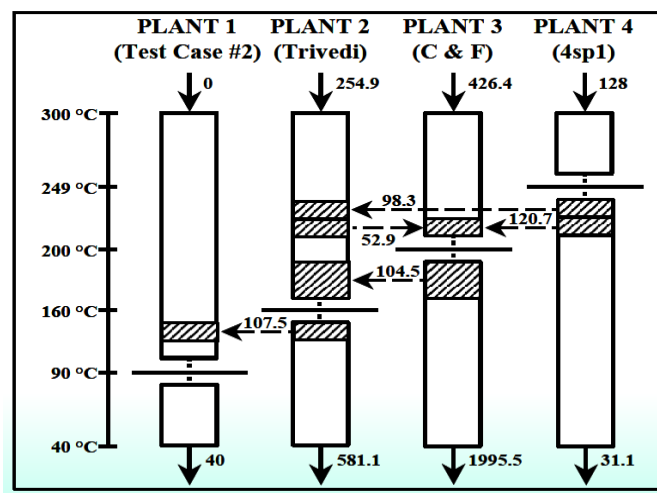
实现中间介质传热需要选择合适的介质。这需要我们建立全厂复合曲线，它包括热阱曲线与热源曲线。

两套装置各自的夹点上方是需要热量的，将 GCC 中夹点上方的曲线（除去 pocket）像 Part2A 中组物流那样组合起来就得到了热阱曲线。类似地，热源曲线就是将 GCC 中夹点下方的曲线（除去 pocket）像 Part2A 中组物流那样组合起来（不过这样得到的热源曲线要镜像翻转一下）。然后两条曲线重叠的部分就是节约的能量。



TOTAL SITE INTEGRATION

不仅是两个装置，3 个，4 个……都可以集成。不过，这时会出现不止一个解决方案，而且 GCC 没法使用了。



HEAT EXCHANGER NETWORKS

换热网络应当使得两套装置不管是否集成都能在最大效率下工作

一旦目标确定，就有程序进行直接集成（流体直接从一个装置流向另一个装置）的设计或者有中间流体（如道氏热载体）的系统

CONCLUSIONS

- 如果一些小问题得到解决（控制、输送液体、开停车的能力），整体现场集成将显示出潜力。
- 虽然两个装置能够节约多少能量可以通过夹点被预测，但是换热网络设计无法使用夹点设计方法
- 数学规划是预估多装置的最小公用量和 HEN 设计的合适工具

Part 3e

Shell & Tube Heat Exchanger Optimal Design

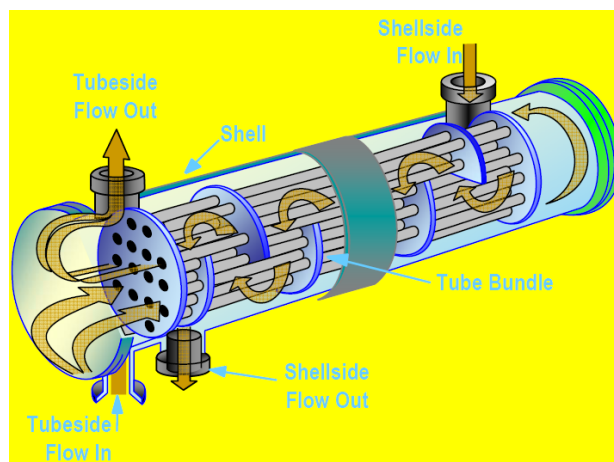
管壳式换热器的最佳设计

TRADITIONAL APPROACH

化工原理中我们学习了换热器的传统设计方法，它是试验 - 验证的方法：

- 首先估计一个传热系数 U
- 得到对应的换热面积
- 根据换热面积提出换热器的几何结构
- 根据几何结构得到传热系数
- 如果 U 大于起初的估计值，那么接受当前设计

如果 U 小于起初的估计值，就要重新设计几何结构直到得到验证



1. Make basic specifications for the shell and tube heat exchanger according

to the nature of the process task: fluid allocation (tube side and shell side), shell and head types, and heat exchanger configuration (e.g.number of tube and shell passes).

2. Pick values for options of tube diameter, tube length, baffle spacing, etc.
3. Evaluate the area using an estimate of the overall heat transfer coefficient

$$U^{\text{est}}$$

$$4. A^{\text{est}} = \frac{\hat{Q}}{U^{\text{est}} \Delta \hat{T}_{\text{lm}} F}$$

\hat{Q} 和 $\Delta \hat{T}_{\text{lm}}$ 是换热器的热负荷以及对数平均温差，由换热任务得到
 F 是对数平均温差的关联系数

5. 在表格中选择管数、管长以及对应兼容的壳直径，使得换热面积接近 A^{est}
6. Calculate the overall heat transfer coefficient (U) using recommended values of fouling factors and evaluate tube-side and shell-side pressure drops.

用推荐的污垢热阻计算总传热系数，估计管程和壳程的压降

7. Compute the heat exchanger area based on the chosen geometry and compare it to the required area calculated through using the evaluated overall heat transfer coefficient from the previous step.

基于选定的换热器几何结构计算实际换热面积，与用先前步骤中算出的总传热系数计算的预估换热面积比较

8. If the overdesign is within an acceptable range and the pressure drops are below the maximum allowable values, then the design is usually finished.

如果设计在合理范围内，且压降低于最大允许值，那么设计完成

9. Otherwise, adjustments to geometrical variables are made based on the analysis of the current trial, aiming at fixing the problems observed and return to Step 5.

否则，基于当前的尝试调整几何参数，回到 5

(上面的都是化原学过的，我就不翻译啦!)

U Value Calculations

One can use

- Kern Method
- Bell Delaware
- Commercial software (HTRI)

Part 3 - F

Simultaneous Design of HEN and Heat Exchangers

同时设计换热网络与换热器

Simultaneous design of HEN+HEX

Problem Statement

Given a set of hot and cold streams as well as hot and cold utilities, obtain the heat exchanger network and the basic design of the heat exchangers in the network.

已知一系列冷热物流与冷热公用，得到换热网络与换热网络中换热器的设计

Classical Sequential Approach 经典顺序方法

Propose values of the heat transfer coefficients $U_{i,j}$ between streams (taken from typical values) 提出传热系数 $U_{i,j}$

Obtain the Heat Exchanger Network 得到换热网络

Use the heat transferred in each match ($q_{i,j}$) and the LMTD i,j to get design the exchanger. 用每个传热单元的传热量 $q_{i,j}$ 和 $LMTD_{i,j}$ 设计换热器

It is not known if the network is the correct one in the first place.

最开始不知道这个网络对不对

Naïve Sequential Approach

Propose values of the heat transfer coefficients $U_{i,j}$ between streams (taken from typical values) 提出传热系数 $U_{i,j}$

Obtain the Heat Exchanger Network

Design all heat exchangers

Use the $U_{i,j}$ of the new exchangers to update their values and design the network again. 用新换热器的 $U_{i,j}$ 更新原来的值，再设计一次换热网络

Stop when there are no new changes 直到没有变化

Simultaneous Approach

Build a model that can obtain both the network and the hex at the same time
建立一个可以同时得到换热网络和换热器的模型

For that we need to merge a code for the networks and add to A i,j $q_{i,j}$ $U_{i,j}$ $F_{i,j}$ $LMTD_{i,j}$

为了这个我们需要合并换热网络的代码，并添加、 $A_{i,j} = q_{i,j} / (U_{i,j} F_{i,j} LMTD_{i,j})$ 、a code that would obtain the $U_{i,j}$ and the $F_{i,j}$ 一个获得 $U_{i,j}$ 和 $F_{i,j}$ 的代码

This has not been done. Instead, an enumeration technique was used

这还没完，还要使用枚举法：

An enumeration of all networks, one by one was set up.

——建立所有的网络

Each network is evaluated for its optimal cost.

每个网络计算一下最优成本

Unlike other procedures, the procedure renders the globally optimal solution
与其他程序不同的是，该程序给出全局最优解

2 example and solutions (见 PPT)

Conclusions

All existing methods (except our) give good results

Methods tends to be trapped in local optimal

None of previous HEN HEX approaches claims for global optimality

Other approaches, like Stochastic Methods, Genetic algorithms, etc do not guarantee optimality

Energy consumption varies from structures obtained using the constant $U_{i,j}$ model

The sizes of the exchangers vary

THE STRUCTURE CAN VARY

所有现有的方法(除了我们的)都给出了良好的结果

方法往往陷入局部最优

以前没有任何 HEN HEX 方法声称具有全局最优性

其他方法，如随机方法，遗传算法等不能保证最优

使用常数 $U_{i,j}$ 模型得到的结构不同，能耗也不同

换热器的规模各不相同

结构可以不同

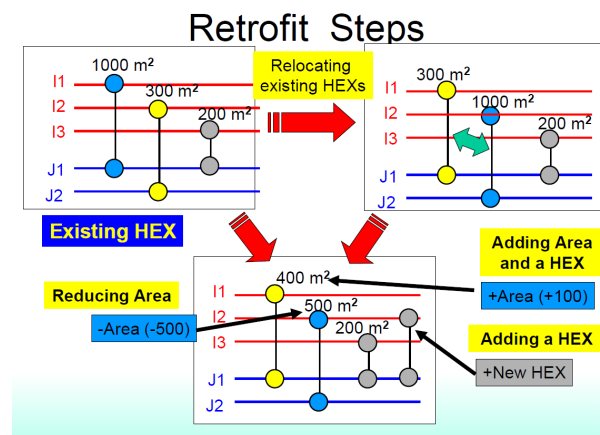
Part 3 - G

Heat Exchanger Network Retrofit

换热网络改造

什么是改造 (Retrofit) ?

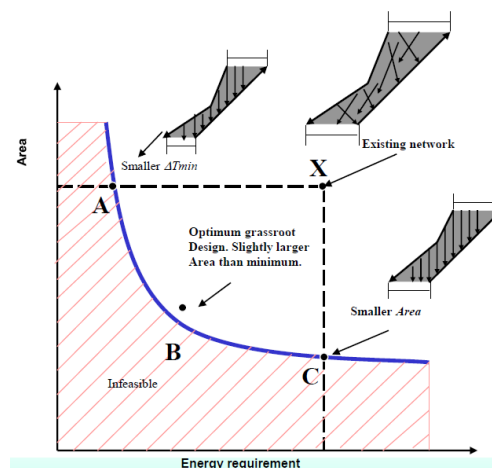
换热网络改造就是给定一个已经存在的换热网络，通过修改（如增加现有换热器换热面积，增加新的热交换器，减少现有的换热器的面积，改变换热器在网络中的位置等）带减少能量消耗，获得更多利润的过程。



虽然通常都是以利润为目标，但环境原因有时也会占主导地位。比如政府发布减少二氧化碳排放的法规可能会迫使减排，而利润导向可能会使排放超标。如果以税收的形式来衡量和支付温室气体排放，问题就会回归到纯粹的利润驱动。上述两种情况的结合表现为“限额与交易”。本课中我们考虑纯粹的利润。

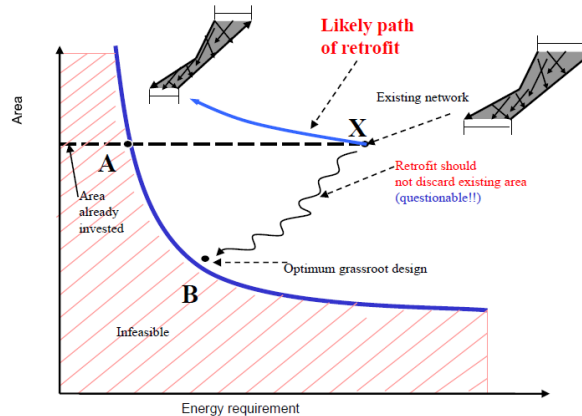
夹点技术目标 Pinch Technology Targeting

这是一张面积 需求能量关系图（后面称为 A-E 图），图中存在一条最优曲线，这代表着面积和能量需求不能同时都低，图上的点代表了一个换热网络的情况，显然点不能在曲线下方

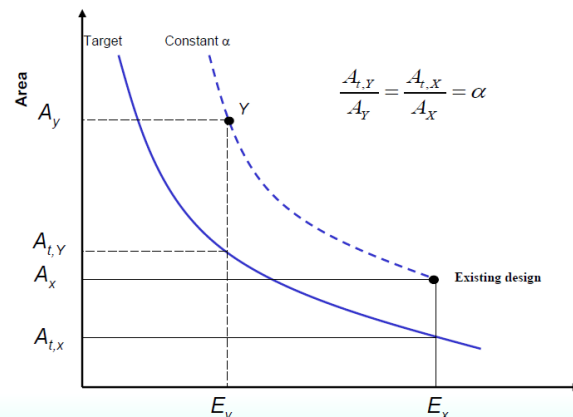


可能的夹点技术路径 PT Likely Path

换热网络改造之后，它在 A-E 图上的点位置自然会变化，形成了一条路径，一般来讲，因为我们的目的是为了减少能耗，所以路径一定是朝左的；可能性最大的路径方向是左上（换热面积增多），也有人认为向左下也是可行的。



TARGETING CONSTANT AREA EFFICIENCY



在虚线上，任意一点对应的能量需求下的最小换热面积与对应的换热面积比值为常数 α ，这个常数称为面积效率（Area Efficiency）

Pinch Technology Procedure

接下来介绍一种应用夹点技术（Pinch Technology）的改造方法。

换热网络改造时,它在 A-E 图上的移动路径一开始是没有定义的,后来它被“规定”为原有的效率常数 α 的等值线。那么,新换热网络面积的确定遵循以下步骤

- ① 从已经存在的系统起手。获得它的理想换热面积 $A_{t,X}$ （假设是垂直换热）
- ② 计算当前换热网络的能耗 E_X
- ③ 选择一个小于 E_X 的 E_Y
- ④ 得到 E_Y 对应的理想换热面积 $A_{t,Y}$ （同样假设垂直换热）
- ⑤ 根据 $\alpha = \frac{A_{t,X}}{A_X} = \frac{A_{t,Y}}{A_Y}$ ，得到“真正”的目标换热面积 A_Y
- ⑥ 得到换热器数量 N_Y （使用最小传热单元数定理）

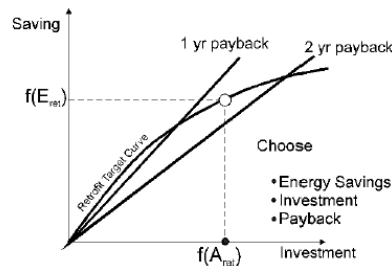
⑦ 计算成本：成本 = 额外面积 $(A_Y - A_X)$ 成本 + 新增换热器 $(N_Y - N_X)$ 成本

⑧ 得到投资年限：投资年限 = 新增投资费用 / 一年的能量节约费用

(投资年限是中文书中的名称，PPT 中的英语为 Payout)

(有时候这个指标会用投资回报率 ROI 和净现值 NPV 代替)

下图是投资与节能关系曲线与投资回收年限曲线，两曲线的交点对应了特定的夹点传热温差。一般来说投资回收年限会先确定下来，然后确定交点，这个交点会对应一个夹点温差。

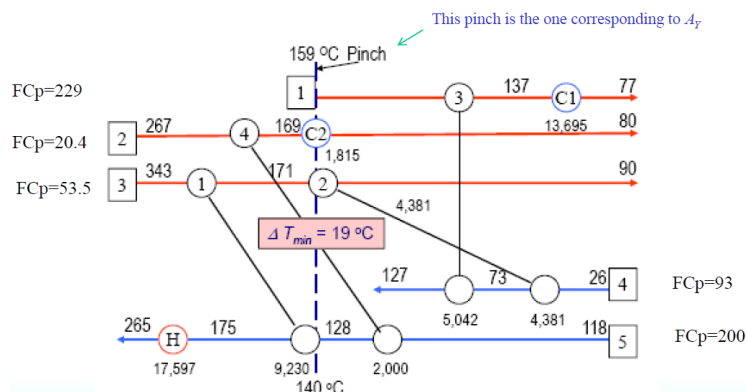


HEN Retrofit

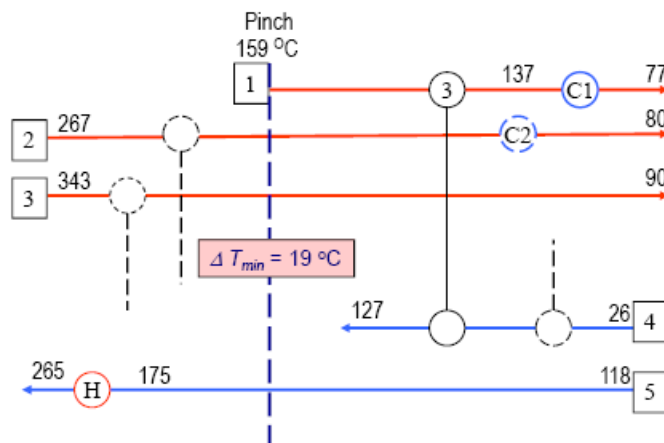
得到了目标夹点温差之后，就可以得到夹点温度，然后进行下一步：

① 找出并删除跨越夹点的换热器

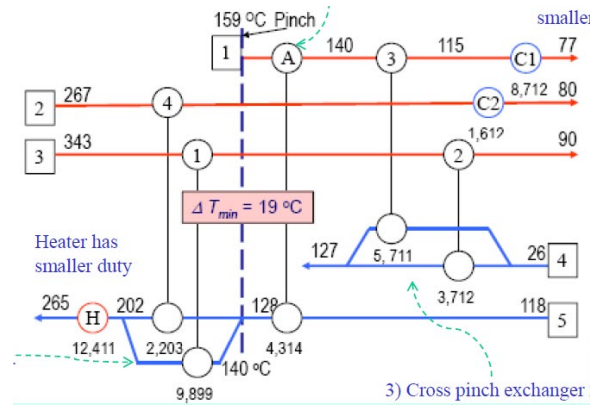
我们来看一个例子。下图中，换热器 1、2 和 4 跨越了夹点传热，冷却器 C2 一部分在夹点上面了，这些都违反了三大铁律，需要删除。



因为是改造，已经存在的换热器要尽可能利用，所以在换热网络图中，我们保留合理的一端，删去不合理的一端。



然后通过一系列手段，修复当前网络的 bug：



- 新增换热器修复了换热器 1 和 4 跨越夹点传热的 bug
- 跨越夹点的换热器 1 和 4 移除后需要分流，这两个换热器也可能要增大面积
- 通过增大换热面积，换热器 2 的 bug 也被修复，但因此也需要分流

② 通过环路与路径分析最小化面积缩减

- 选择 EMAT 的值
- 增加环路中换热器的负荷，直到一个或多个换热器成为瓶颈 (bottleneck)。

注：应该指的是传热温差低于 EMAT 了

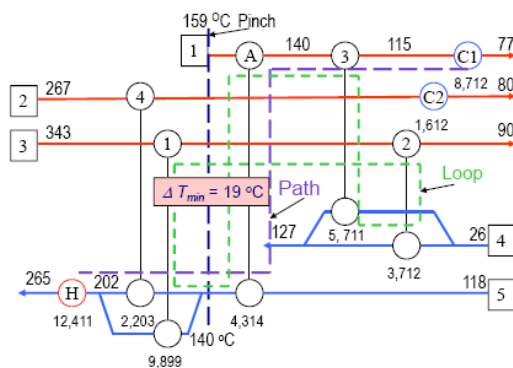
- 如果没有实现全部的节约潜能，尝试通过以下方式消除瓶颈：

1 修改支路分流的分配比例

2 如果一个换热器的推动力还比较大，而另一个已经达到了瓶颈就重新排列它们或加入分流，使推动力均匀一些。

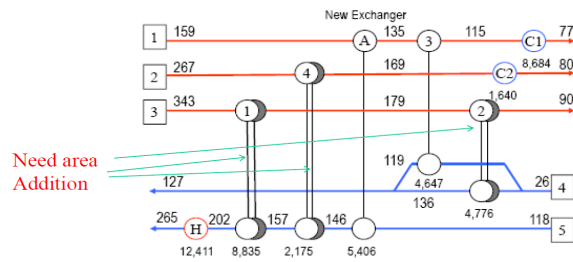
- 在环路中再次增加换热器的负荷，直到实现了全部的节约潜能。
- 利用热负荷回路重新分配网络中的推动力，从而减少总体面积需求。
- 尝试使用路径来进行能量放松 (relaxations)，同时调整换热面积。

比如说，我们在之前的换热网络中找出一个回路：



该回路用于转移热负荷，使换热器 1 和 4 不需要额外增加换热面积。我们将换热器 1 的负荷减少 X，并相应调整其他所有的负荷。当面积足够大时，得到最佳 X。4 和 3 将相应地改变其负荷。

最后得到的换热网络如下，1，2，4 都新增了换热面积



CONCLUSIONS for PDM Method

- 目标程序是基于不确定的经济预测
- 通过检查做出更改
- 循环和路径分析需要建立专业知识
- 不是完全系统的，也不是自动的
- 可以很好地解决小问题
- 大问题的组合选择可能会增加。

接下来介绍一种系统的 & 自动的数学规划方法

Use of Mathematical Programming

关于问题的一些分析：

目标函数 (Objective Function)：Maximize {NPV of Savings (\$) Cost}

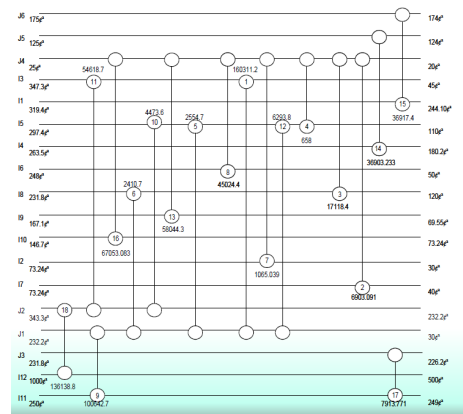
节约的费用：公用工程的费用 (现有的 新的)

新增成本：新换热器，新的面积，改动位置费用等

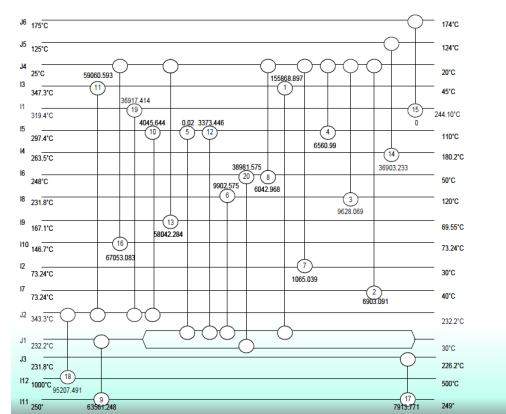
增加的换热面积越大，费用越高，可以节约的能量就越多，这里存在权衡。

我们将使用 MILP 运输模型，这是一个线性模型，它考虑：

- 换热器的增加
- 通过新的壳程增加的换热面积
- 通过管子的全新排列增大的换热面积
- 位置调整与 re piping



Before

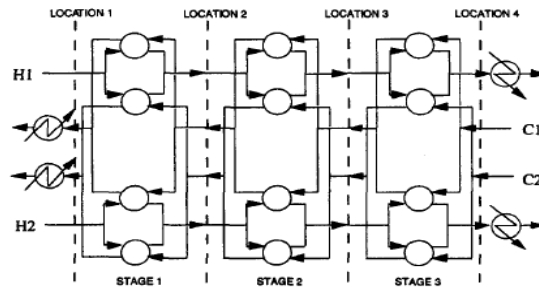


After

HE	Original Load MJ/hr	Retrofit Load MJ/hr	Original Area m2	Retrofit Area m2	Area Addition m2	Shell Addition	Cost \$
1	160,311.20	155,868.90	4,303.20	3,926.25			
2	6,903.09	6,903.09	59.40	63.80	4.40		342.03
3	17,118.40	9,628.07	33.40	21.53			
4	658.00	6,560.99	2.30	16.63	14.33	YES	6,406.84
5	2,554.70	0.02	26.30	28.93	2.63		204.58
6	2,410.70	9,902.58	24.60	398.53	373.93	YES	34,379.01
7	1,065.04	1,065.04	5.50	5.87	0.37		28.70
8	45,024.40	6,042.97	145.00	41.66			
9	100,642.70	63,561.25	1,212.70	962.01			
10	4,473.60	4,045.64	93.70	93.70			
11	54,618.70	59,060.59	685.70	1,239.90	554.20	YES	48,402.09
12	6,293.80	3,373.45	40.00	44.00	4.00		311.15
13	58,044.30	58,042.28	183.30	182.39			
14	36,903.20	36,903.23	101.60	101.47			
15	36,917.40	0	93.90	0			
16	67,053.08	67,053.08	278.10	288.97	10.87		845.32
17	7,913.77	7,913.77	53.50	52.24			
18	136,138.80	95,207.49	976.40	709.00			
19	36,917.41			727.96			61,918.53
20	38,981.58			651.93			56,004.54
			8,318.60	9,556.76	14.88%	3	208,842.80

Costs	Existing	Retrofitted
Total utility cost	\$ 6,865,616/yr	\$ 5,004,800/y
Total fixed and area cost		\$ 208,842/yr
Total cost	\$ 6,865,616/yr	\$ 5,213,643/y
Savings		-24%

采用 3-B 中的 Stage Model 来处理面积增加。如果不引入分流：



引入变量： $z(i,j,k)=1$ 表示这个位置存在一台换热器

同时还要在已有阶段中留出空的阶以便直接添加新的换热器

一台换热器的费用由已有换热器增加的面积或整个面积（如果是新换热器的话）给出

如果引入分流：

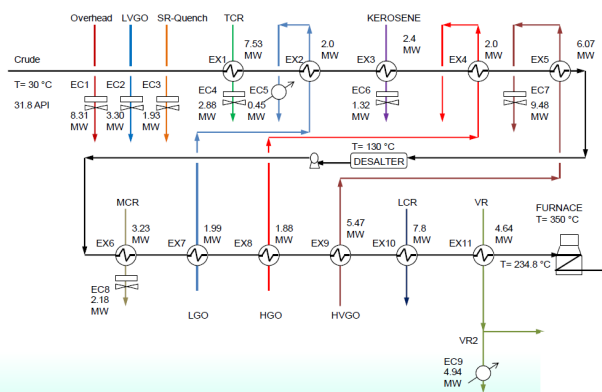
引入条件：当物流 i 和 j 在 k 阶有换热器换热时， $\sum\{k,z(i,j,k)\} \geq 1$

这允许“串联”的网络允许将换热器重新排列为平行甚至改变顺序

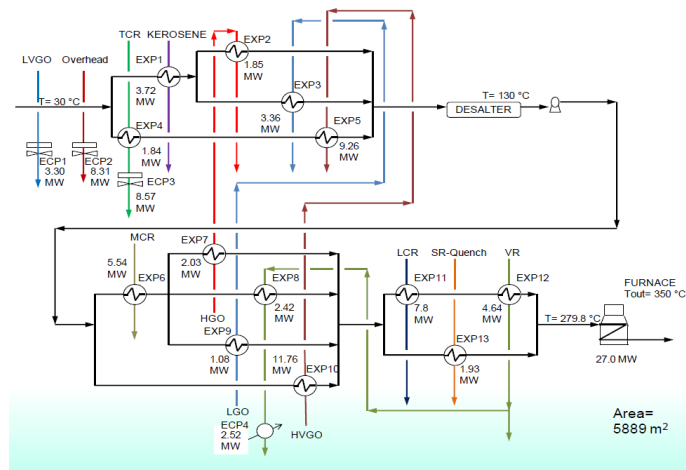
同样还要在已有阶段中留出空的阶以便直接添加新的换热器

同样一台换热器的费用由已有换热器增加的面积或整个面积（新换热器）给出

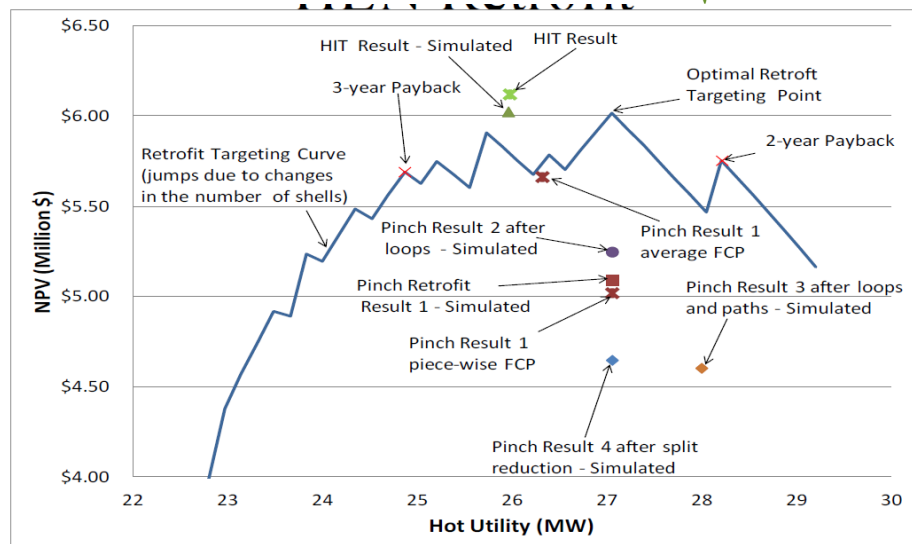
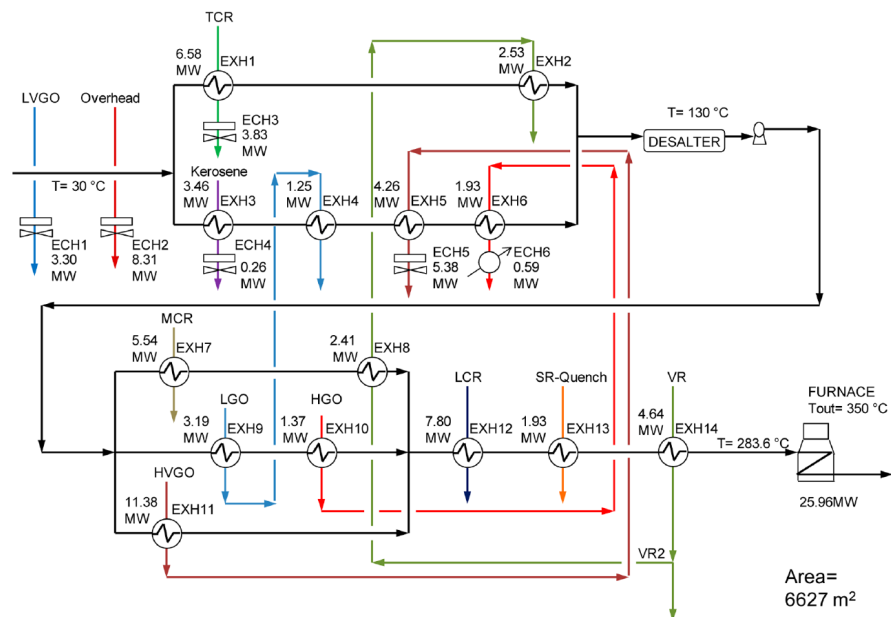
我们看一个例子：



使用 PDM 改造后：



使用 HIT 改造后 (变 Cp)



NPV/热公用图,

- 夹点设计方法并不总能可靠地得到好的改造
- 阶段模型 (Stage Model) 当需要分流的时候又有不足的

- The stages/substages model 很有前途，但没人尝试过
- 有人尝试了其它的方法（GA, PSO 等）
- HIT 方法已经被证明成功并得到工业应用
- 改造与 HEX 设计同时进行的挑战尚未得到主流验证

Synheat Use for Minimal (最小的) Changes

我们提出修改 Synheat 模型，使得我们能够只通过增减换热面积进行改造

① 修正所有 $z(i,j,k)$ 的值

$z.fx(i,j,k) = 0;$

$z.fx('i1','j2','k1') = 0;$

对所有已有换热器重复以上指令

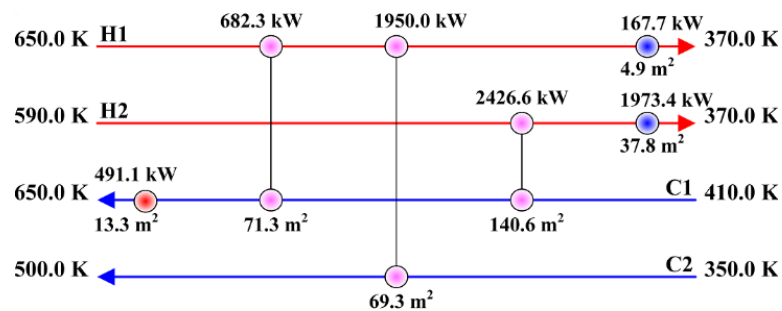
② 将能量消耗修正为新的值

$sum(j,qh(j)) = E_{new};$ （这是一条要加入的 equation）

③ 对得到的每个答案计算加上或减去的面积

④ 计算投资回报

假设有下面这个例子



假设现在的状况是

--- 208 PARAMETER a

1 2

1.1 53.525

1.2 102.084

2.1 67.662

---- 208 VARIABLE qc.L

2 2370.000

---- 217 PARAMETER acu area coolers

2 42.307

---- 208 VARIABLE q.L

1 2

1.1 850.000

1.2 1950.000

2.1 2030.000

---- 208 VARIABLE qh.L

1 720.000

---- 217 PARAMETER ahu area heaters

1 17.218

新的答案是

--- 208 PARAMETER a

1 2

1.1 63.625

1.2 102.084

2.1 75.429

--- 208 VARIABLE qc.L

--- 208 VARIABLE q.L

1 2

1.1 850.000

1.2 1950.000

2.1 2100.000

--- 208 VARIABLE qh.L

2 2300.000 1 650.000 *Existing plant was 720*
 --- 217 PARAMETER acu area coolers --- 217 PARAMETER ahu area heaters
 2 41.55 1 16.103

假设新的换热面积是一个新的 shell

Original area		New Area	
1	2	1	2
1.1	53.525	1.1	63.625
1.2	102.084	1.2	102.084
2.1	67.662	2.1	75.429

- 计算面积差 $\Delta a(i,j,k)$
- 计算改造费用: $RC = N_{\text{new}}c_f + d \text{ Sum}((i,j,k), \Delta a(i,j,k))^e$

N_{new} 是新 shell 的数量; 忽略面积减少的费用

- 计算能量节约 $ES = E_{\text{exist}} - E_{\text{new}}$
- 计算用水节约 $WS = W_{\text{exist}} - W_{\text{new}}$
- 计算改造后一年能节约的费用 $RS = c_Q ES + c_w WS$
- 计算投资回报 $ROI = RS/RC$
- 重复以确定最佳 ROI

使用 Stages Model 改造换热网络

我们的例子中, 只有换热面积发生了变化

