



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV PROCESNÍHO A EKOLOGICKÉHO
INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF PROCESS AND ENVIRONMENTAL
ENGINEERING

APLIKACE MODERNÍCH METOD SYNTÉZY SÍTĚ VÝMĚNY TEPLA

APPLICATION OF RECENT METHODS FOR SYNTHESIS OF HEAT EXCHANGER NETWORK

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VLASTIMIL KUNC

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ZDENĚK JEGLA, Ph.D.

BRNO 2009

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav procesního a ekologického inženýrství

Akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Vlastimil Kunc

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Procesní inženýrství (3909T003)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Aplikace moderních metod syntézy sítě výměny tepla

v anglickém jazyce:

Application of recent methods for synthesis of heat exchanger network

Stručná charakteristika problematiky úkolu:

1. Seznámit se s metodou

Cíle diplomové práce:

Seznámit se s metodami pro syntézu sítě výměny tepla a sestavit programovou realizaci deterministické výpočtové metody. Na praktických průmyslových úlohách ukázat možnosti, úskalí, výhody a nevýhody současných metod syntézy výměňkových sítí formou srovnání metody

Seznam odborné literatury:

1. Stehlík, P.: Integrace procesů a její význam pro redukcii spotřeby energie a škodlivých emisí -základní principy, Nakladatelství "Procesní inženýrství", edice MAPRINT, Praha (1995).
2. Linnhoff, B., et al.: User Guide on Process Integration for the Efficient Use of Energy, IChemE, Rugby, U.K. , latest updated reprint (1994).
3. Smith, R.: Chemical Process Design, McGraw-Hill (1995).
4. Errico, M., et al.: A deterministic algorithm for the synthesis of maximum energy recovery heat exchanger network, Computers and Chemical Engineering 31, pp.773–781 (2007).

Vedoucí diplomové práce: doc. Ing. Zdeněk Jegla, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2008/2009.

V Brně, dne 18.11.2008

L.S.

prof. Ing. Petr Stehlík, CSc.
Ředitel ústavu

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Diplomová práce se zabývá problematikou syntézy sítě výměny tepla a srovnává současné metody, s důrazem na *Pinch Design Method* a deterministickou metodu. Pomocí softwaru *Maple* je vytvořen program pro návrh sítě výměny tepla deterministickou metodou. Tato metoda je aplikována na několika konkrétních příkladech.

Klíčová slova

Síť výměny tepla, výměník, *Pinch Design Method*, deterministický model, opakovací jednotka, externí energetické zdroje, maximální procesní výměna tepla

Abstract

Master's thesis deals with the problems of heat exchanger network synthesis and compare the present methods with emphasis on *Pinch Design Method* and deterministic method. Based on theoretical formulation of deterministic model the computer program for heat exchanger network synthesis was developed in the software *Maple* environment. Developed software implementation of deterministic method has been applied to several case studies.

Key words

Heat exchanger network, exchanger, *Pinch Design Method*, deterministic model, repetitive unit, utility, maximum energy recovery

Bibliografická citace

KUNC, V. *Aplikace moderních metod syntézy sítě výměny tepla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009. 55 s. Vedoucí diplomové práce doc. Ing. Zdeněk Jegla, Ph.D.

Čestné prohlášení

Prohlašuji, že tuto diplomovou práci jsem napsal sám za pomoci mého vedoucího diplomové práce doc. Ing. Zdeňka Jegly, Ph.D., a za pomoci literatury uvedené v seznamu použitých zdrojů.

V Brně dne 26. 5. 2009

Bc. Vlastimil Kunc

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Zdeňku Jeglovi, Ph.D. za jeho odborné vedení a cenné rady, které mi byly během tvorby diplomové práce prospěšné. Dále bych chtěl poděkovat ing. Vojtěchu Turkovi za poskytnuté konzultace v oblasti matematického programování.

Obsah

Seznam použitých symbolů	8
1 Úvod	10
1.1 Stručný přehled historického vývoje metod syntézy sítě výměny tepla.....	10
1.1.1 Heuristické metody.....	10
1.1.2 <i>Pinch Design Method</i>	10
1.1.3 Matematické programování.....	11
1.1.4 Evoluční metody (Genetické algoritmy)	12
2 Základy <i>Pinch Design Method</i>	13
2.1 Kompozitní křivky.....	14
2.2 Pravidla <i>Pinch Design Method</i>	14
3 Ukázka aplikace metody PDM.....	16
3.1 Řešení oblasti nad PINCHEm.....	17
3.2 Řešení oblasti pod PINCHEm.....	19
4 Další možnosti <i>Pinch Design Method</i>	21
5 Principy deterministické metody	25
6 Zpřesněná formulace deterministické metody a její automatizace.....	26
6.1 Účelová funkce	26
6.2 Omezení ve formě rovnosti	26
6.3 Logická omezení.....	27
6.4 Omezení ve formě nerovnosti.....	27
6.5 Ilustrativní ukázka konkrétní formulace jednotlivých omezení	28
7 Ukázka aplikace deterministické metody	30
8 Řešení praktických příkladů	32
8.1 Příklad 1.....	32
8.2 Příklad 2.....	34
8.3 Příklad 3.....	36
9 Závěr.....	42
Seznam použitých zdrojů.....	43
Příloha 1.....	44
Příloha 2.....	48
Příloha 3.....	50
Příloha 4.....	52
Příloha 5.....	54

Seznam použitých symbolů

Symbol	Jednotka	Význam
A	[m ²]	plocha výměny tepla
CP _C	[kW/°C]	tepelná kapacita chladného proudu
CP _H	[kW/°C]	tepelná kapacita horkého proudu
G	[W]	maximální dovolený výkon jednotlivých výměníků
H	[W]	tepelný obsah (výkon)
K	[W/(m ² .K)]	součinitel prostupu tepla
M	[°C]	manipulační konstanta teplotního rozdílu
N _{max}	[-]	maximální počet výměníků tepla v síti
n _C	[-]	počet chladných proudů
n _H	[-]	počet horkých proudů
Q	[W]	celkový přenos tepelného výkonu výměníku
Q _C	[W]	výkon chladného servisu
Q _H	[W]	výkon horkého servisu
T	[°C]	teplota
T ₁₁	[°C]	vstupní teplota teplejšího proudu
T ₁₂	[°C]	výstupní teplota teplejšího proudu
T ₂₁	[°C]	vstupní teplota chladnějšího proudu
T ₂₂	[°C]	výstupní teplota chladnějšího proudu
T _{C,in}	[°C]	vstupní teplota chladného proudu (<i>Pinch Design Method</i>)
T _C ^s	[°C]	vstupní teplota chladného proudu (Deterministický model)
T _{C,out}	[°C]	výstupní teplota chladného proudu (<i>Pinch Design Method</i>)
T _C ^t	[°C]	výstupní teplota chladného proudu (Deterministický model)

$T_{\text{cold,in}}$	[°C]	vstupní teplota na chladném konci výměníku
$T_{\text{cold,out}}$	[°C]	výstupní teplota na chladném konci výměníku
$T_{\text{H,in}}$	[°C]	vstupní teplota horkého proudu (<i>Pinch Design Method</i>)
T_{H}^{s}	[°C]	vstupní teplota horkého proudu (Deterministický model)
$T_{\text{H,out}}$	[°C]	výstupní teplota horkého proudu (<i>Pinch Design Method</i>)
T_{H}'	[°C]	výstupní teplota horkého proudu (Deterministický model)
$T_{\text{hot,in}}$	[°C]	vstupní teplota na horkém konci výměníku
$T_{\text{hot,out}}$	[°C]	výstupní teplota na horkém konci výměníku
$T_{\text{PINCH,C}}$	[°C]	teplot chladných proudů v PINCHi
$T_{\text{PINCH,H}}$	[°C]	teplot horkých proudů v PINCHi
Z	[-]	účelová funkce
α_1, α_2	[W/(m ² .K)]	součinitel přestupu tepla
$\Delta \bar{T}_{\text{in}}$	[°C]	střední logaritmický teplotní rozdíl
ΔT_{min}	[°C]	minimální dovolené teplotní přiblížení
Δ	[-]	binární proměnná označující přítomnost daného výměníku
E	[W]	minimální dovolený výkon jednotlivých výměníků

1 Úvod

Sít výměníků tepla je součástí mnoha výrobních procesů v různých průmyslových odvětvích. Při návrhu procesních linek je snahou dosáhnout ekonomicky hospodárné provedení výrobní linky, tj. dosáhnout co největších úspor energie, popř. redukci emisí. Řešení sítě výměny tepla se skládá ze tří základních fází. První fáze se nazývá targeting, což je předběžná technicko-ekonomická rozvaha za účelem určení ekonomicky optimálního stupně výměny tepla mezi procesními proudy. Důležitým výsledkem této fáze je, mimo jiné, zejména nalezení minimálního dovoleného teplotního přiblížení (ΔT_{\min} [°C]). Na targeting navazuje druhá fáze, často označovaná jako syntéza. Úkolem fáze syntézy je nalezení optimálního osazení sítě výměny tepla jednotlivými výměníky. Třetí fází je detailní návrh jednotlivých výměníků tepla ve výměňkové síti.

Tato práce je zaměřena na problematiku řešení druhé fáze, tedy fáze syntézy. Mezi doposud nejrozšířenější metody při řešení výměňkových sítí patří metoda *Pinch Design Method* (PDM), dříve nazývaná *Pinch Analysis* či *Pinch Technology* [1]. V současné době nastupují a rychle se rozšiřují metody numerické. Z řady modelů publikovaných v této oblasti se v poslední době jako progresivní jeví tzv. deterministický model. Tento numerický model bude v této diplomové práci podrobně popsán a srovnány jeho vlastnosti a možnosti s možnostmi a vlastnostmi *Pinch Design Method* dosud stále velmi oblíbené mezi projektanty.

1.1 Stručný přehled historického vývoje metod syntézy sítě výměny tepla

Přesto, že problematika řešení fáze syntézy výměňkových sítí je poměrně mladým odvětvím (je řešena přibližně v posledních padesáti letech), prošla poměrně významným vývojem. V následujících odstavcích je proto proveden stručný přehled vývoje v této oblasti.

1.1.1 Heuristické metody

Jsou to vývojově prvotní metody navrhování výměňkové sítě. Spadají do počátků řešení fáze syntézy (60. až 70. léta 20. století). Tyto metody se opírají o tzv. heuristická pravidla, tj. soubor prakticky ověřených poznatků opírajících se o termodynamiku. Tyto pravidla vychází z praktických zkušeností, platí tedy přibližně nebo mají intuitivní charakter, obvykle se tyto pravidla nedají rigorózně matematicky dokázat. Tyto metody jsou jednoduché, avšak u výsledného řešení nelze zaručit, že je pro daný účel optimální.

1.1.2 *Pinch Design Method*

Pinch Design Method (PDM) byla vytvořena v 70. letech 20. století. Je to relativně jednoduchá, srozumitelná a velmi oblíbená metoda, při úlohách velkého rozsahu je však značně zdlouhavá. Je založena na termodynamických a ekonomických principech a na existenci bodu zvaného PINCH. Možnosti této metody pokrývají jak oblast projektování

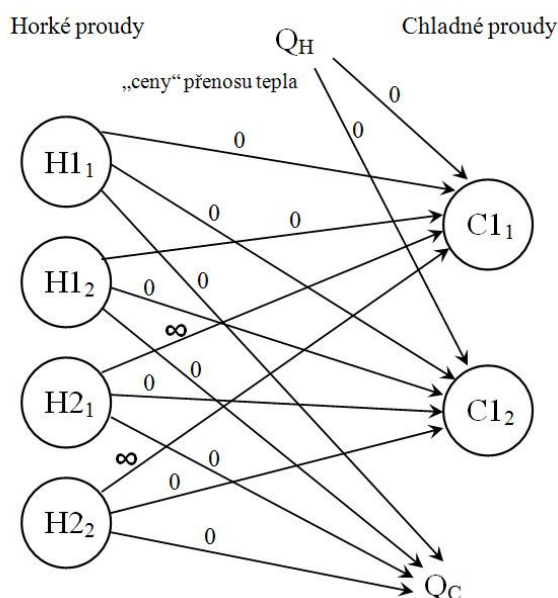
nových technologických linek, tak oblast rekonstrukce. Blíže jsou této metodě věnovány kapitoly 2 až 4 diplomové práce.

1.1.3 Matematické programování

Počátky použití metod matematického programování na případy syntézy výměňkových sítí spadají do období prvních publikací o *Pinch Design Method*. Metody matematického programování se snaží v podstatě kombinatorický problém syntézy výměňkové sítě řešit systematickým způsobem využitím optimalizačních algoritmů a dosáhnout tak globálního optima, jehož dosažení není aplikací *Pinch Design Method* zaručeno. Dále jsou stručně představeni představitelé nejvýznamnějších matematických modelů používaných v dosavadním průběhu vývoje metod matematického programování v oblasti syntézy výměňkové sítě.

Transportation model („transportní“ model)

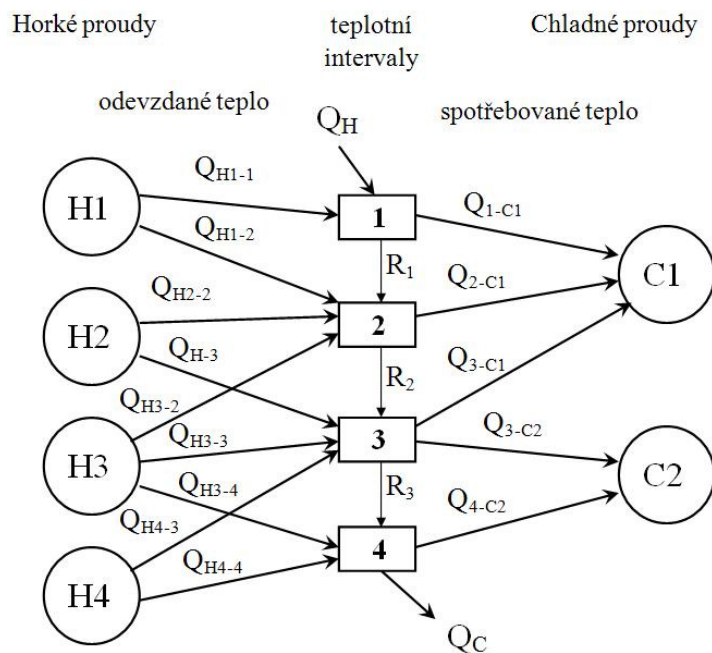
Transportation model je prvním modelem matematického programování (r. 1983) aplikovaným na fázi syntézy výměňkové sítě. Je to analogie s prvními algoritmy operačního výzkumu – nalezení nejlevnější cesty (dopravy, transportu). Později byl nahrazen „překládkovým“ modelem. Princip „transportní“ modelu je na *Obr. 1*.



Obr. 1 Princip „transportního“ modelu

Transshipment model („překládkový“ model)

Využívá principy PDM pro stanovení teplotních intervalů, ve kterých se tepelný tok horkých proudů „překládá“ do chladných proudů. Tento model také umožňuje dělení proudů. Princip „překládkového“ modelu je na *Obr. 2*.



Obr. 2 Princip „překládkového“ modelu

Block Decomposition model

Tento model vychází z pravidel PDM. Jeho snahou je dosažení vertikálního přenosu tepla, avšak eliminací dosažení tzv. „spaghetti network“, tj. eliminací sítě s nepřipustně vysokým dělením proudů. Toho je dosaženo tak, že kompozitní křivky používané v PDM jsou rozděleny do jednodušších sekcí (bloků) a v těchto sekcích jsou podle určitých pravidel „vyhlazeny“ tak, aby bylo možno dosáhnout výměny tepla v této sekci co nejjednodušším zapojením výměníků tepla.

Deterministický model

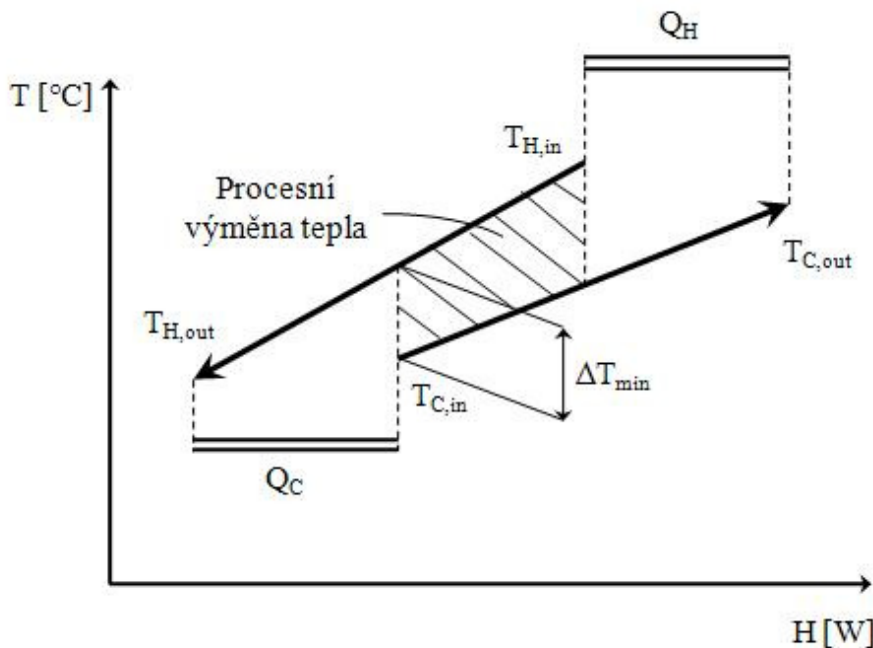
Deterministický model je založen na principu vícenásobného použití tzv. opakovací jednotky. Opakovací jednotka představuje nejjednodušší provedení výměňkové sítě pro daný případ. Vícenásobným použitím opakovací jednotky se vytvoří výchozí modelové schéma (*superstructure*), na které se aplikuje optimalizační algoritmus. Jelikož deterministický model je formulován jako MILP (*mixed integer linear programming*) problém, umožňuje nalezení globálního optima. Na tento v současnosti progresivní model je zaměřeno řešení diplomové práce (viz kapitoly 5 až 8).

1.1.4 Evoluční metody (Genetické algoritmy)

Na problematiku syntézy výměňkové sítě byly ojediněle použity též některé algoritmy z oblasti evolučních metod, zejména genetický algoritmus. Genetický algoritmus je heuristická stochastická optimalizační metoda založená na populačním prohledávání stavového prostoru. Jedná se o výpočtově i časově relativně náročnou metodu, která však nenašla v oblasti výměňkových sítí výraznějšího rozšíření.

2 Základy Pinch Design Method

Základem *Pinch Design Method* je termodynamická analýza a významnou roli zde zaujímá diagram závislosti teploty T [°C] na tepelném obsahu (výkonu) H [W]. Uvažuje se procesní výměnu tepla mezi dvěma proudy. Horký proud je třeba ochladit z teploty $T_{H,in}$ [°C] na teplotu $T_{H,out}$ [°C] a chladný proud ohřát z teploty $T_{C,in}$ [°C] na teplotu $T_{C,out}$ [°C] (*Obr. 3*), kde $T_{H,in}$ a $T_{H,out}$ je vstupní resp. výstupní teplota horkého proudu a $T_{C,in}$ a $T_{C,out}$ je vstupní resp. výstupní teplota chladného proudu.

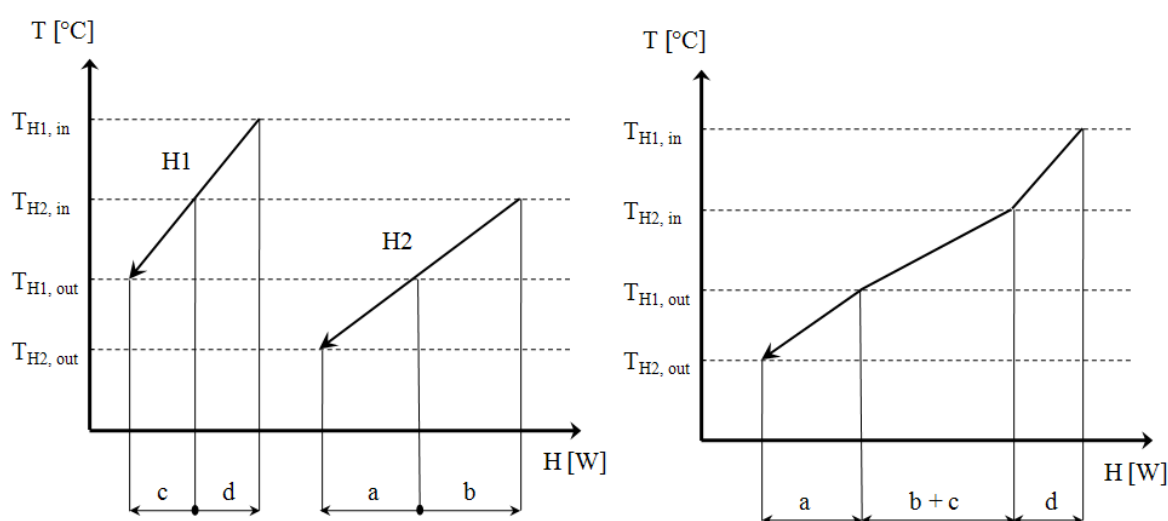


Obr. 3 Diagram „teplota – tepelný obsah (výkon)“ – procesní výměna tepla mezi dvěma proudy

Tyto teploty jsou dány technologickými požadavky procesu, proto nelze křivkami pohybovat ve vertikálním směru. Křivkami lze pohybovat jen v horizontálním směru a poloha je fixována minimálním dovoleným teplotním přiblížením ΔT_{min} . Tato hodnota významně ovlivňuje energetické a investiční náklady. V místech, kde se křivky překrývají, dochází k procesní výměně tepla mezi horkým a chladným proudem. Z *Obr. 3* je zřejmé, že chladný proud je třeba dohřát na teplotu $T_{C,out}$ pomocí horkého servisu (Q_H [W]) a horký proud ochladit na teplotu $T_{H,out}$ pomocí chladného servisu (Q_C [W]). Kolik tepla se vymění procesní výměnou tepla a kolik tepla je třeba přivést či odvést pomocí horkého resp. chladného servisu je jednoznačně určeno hodnotou ΔT_{min} .

2.1 Kompozitní křivky

Doposud byly uvažovány dva procesní proudy. V praxi se lze setkat s více proudy, u složitějších procesů i několika desítkami. Z procesních dat lze sestavit křivky všech proudů v diagramu T-H. Naším cílem je však získat jednu křivku reprezentující všechny horké proudy a jednu křivku reprezentující všechny chladné proudy. Koncové a počáteční teploty jednotlivých proudů nám specifikují teplotní intervaly. V určitých úsecích některých intervalů však může figurovat více proudů. Sestrojení křivky s takto zadanými teplotami je zřejmé z Obr. 4.



Obr. 4 Kompozitní křivka pro horké proudy – postup sestavení

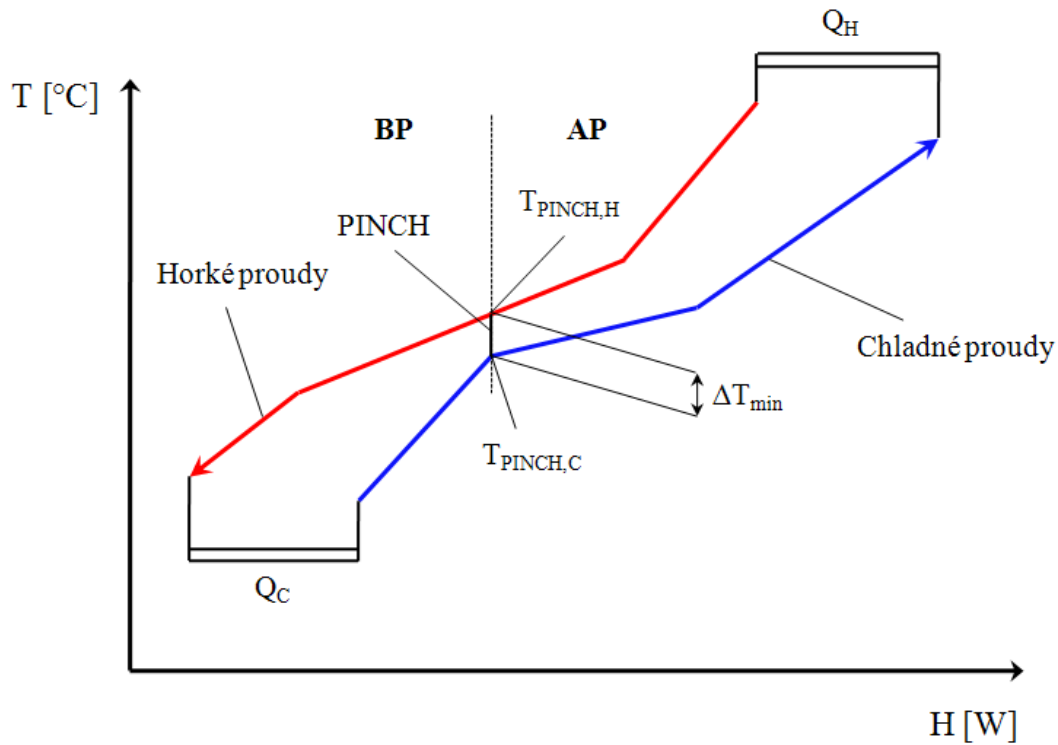
Jsou zde dva horké proudy, jejichž teplotní intervaly se částečně překrývají. Jedná se o intervaly b a c , na nichž lze sečíst jejich entalpie a získat tak část křivky reprezentující oba tyto intervaly. Analogicky se postupuje v případě chladných proudů. Celý tento postup lze zobecnit na jakkoliv složitý systém. Tím se získají křivky reprezentující všechny horké a všechny chladné proudy, tzv. **kompozitní křivky**.

2.2 Pravidla Pinch Design Method

Z Obr. 5 lze vidět, že v určitém místě je vertikální vzdálenost kompozitních křivek minimální a je dána hodnotou minimálního dovoleného teplotního přiblížení ΔT_{min} . Toto místo se nazývá PINCH (odtud název *Pinch Design Method*). Poloha kompozitních křivek fixována PINCHEm udává minimální hodnotu výkonů horkého servisu Q_H [W] a chladného servisu Q_C [W] a hodnoty teplot horkých ($T_{PINCH,H}$ [°C]) a chladných ($T_{PINCH,C}$ [°C]) proudů v PINCHi.

PINCH rozděluje celý systém na dva subsystémy (viz Obr. 5):

- nad PINCHem (*Above the PINCH* - AP)
- pod PINCHem (*Below the PINCH* - BP)



Obr. 5 Kompozitní křivky

Mezi těmito dvěma subsystémy by se nemělo přenášet teplo. Odtud plynou základní pravidla *Pinch Design Method* [1]:

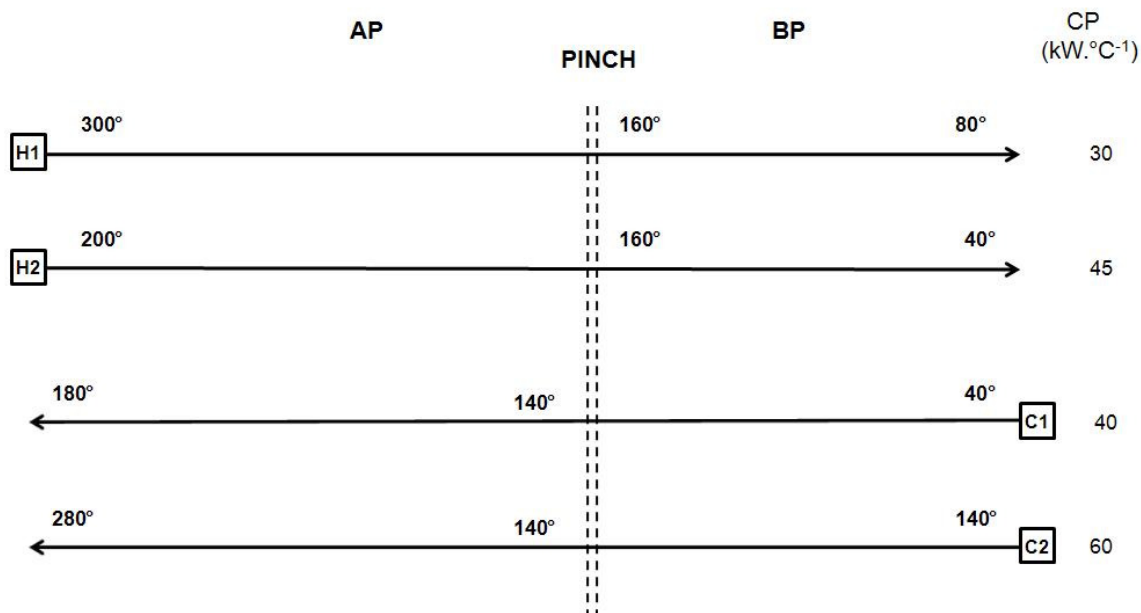
- Nepřenášet procesní teplo přes PINCH (tedy ze subsystému AP do subsystému BP)
- Nepoužívat chladný servis nad PINCHem
- Nepoužívat horký servis pod PINCHem

3 Ukázka aplikace metody PDM

V této kapitole bude popsán postup pro řešení sítě výměny tepla pomocí *Pinch Design Method* na příkladu z článku [6] (*Case1*). Ve fázi *targeting* byla zjištěna teplota minimálního teplotního přiblížení 20°C a intervalová teplota v místě PINCH 150°C V *Tab. 1* jsou zadané hodnoty pro tento příklad, kde CP jsou tepelné kapacity, T_{in} vstupní teploty a T_{out} výstupní teploty jednotlivých proudů. Zadání příkladu lze také znázornit pomocí mřížkové reprezentace (*Obr. 6*).

proud	CP [kW/°C]	T_{in} [°C]	T_{out} [°C]
H1	30	300	80
H2	45	200	40
C1	40	40	180
C2	60	140	280
$\Delta T_{min} = 20\text{ °C}$			

Tab. 1 Vstupní hodnoty k příkladu [6]



Obr. 6 Mřížková reprezentace k příkladu [6]

Pro řešení sítě výměny tepla pomocí PDM jsou dána některá základní pravidla:

- Při osazování sítě jednotlivými výměníky vždy postupovat od PINCHE směrem „ven“
- V oblasti nad PINCHEm začít spojovat proudy, pro které platí:

$$CP_H \leq CP_C \quad (1)$$

- V oblasti pod PINCHEm začít spojovat proudy, pro které platí:

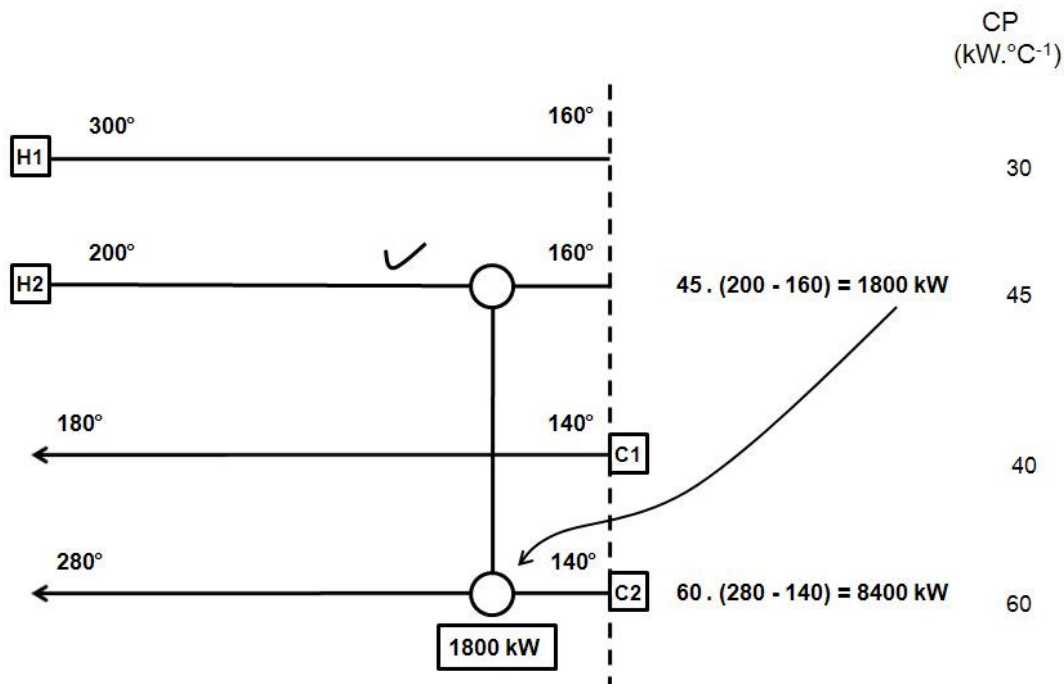
$$CP_H \geq CP_C \quad (2)$$

3.1 Řešení oblasti nad PINCHEm

Při řešení oblasti nad PINCHEm se tedy podle pravidel propojí proudy, kde tepelná kapacita horkého proudu je menší nebo rovna než tepelná kapacita chladného proud. Pomocí teplotního rozdílu a tepelných kapacit se vypočítá výkon umístěného výměníku:

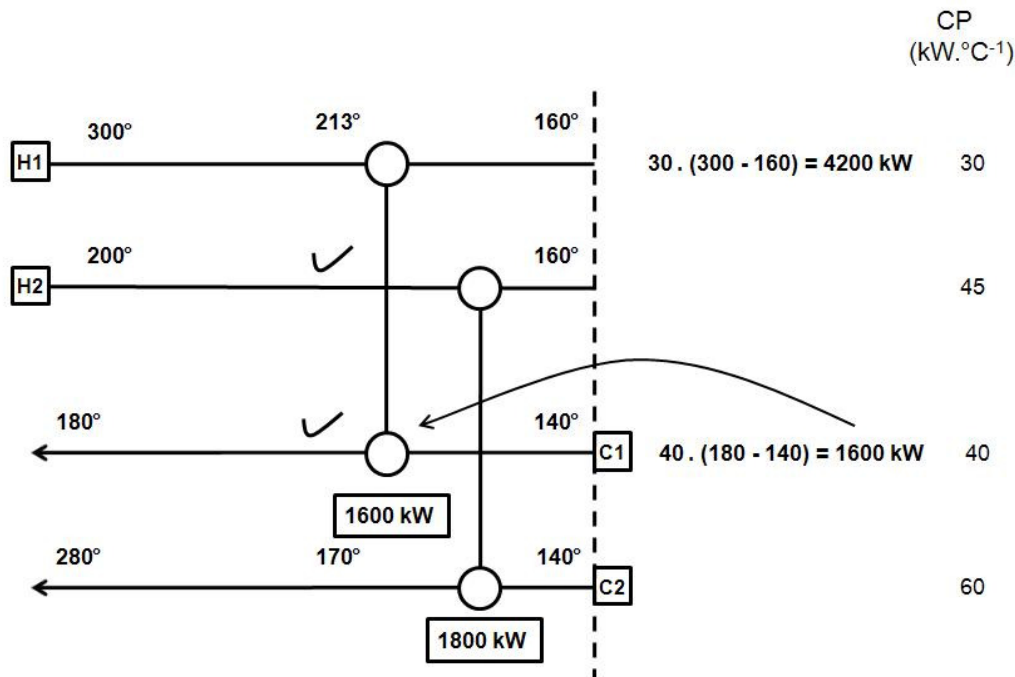
$$\Delta T \cdot CP = Q, \quad (3)$$

kde ΔT [$^{\circ}\text{C}$] je teplotní rozdíl, CP [$\text{kW}/^{\circ}\text{C}$] je tepelná kapacita a Q [kW] je výkon. Jak je vidět z Obr. 7 na horkém proudu H2 vyšel výkon 1800 kW a na chladném proudu C2 vyšel výkon 8400 kW, proto umístěný výměník bude mít výkon 1800 kW a je tedy zcela pokryt teplotní interval horkého proudu H2 v oblasti nad PINCHEm.



Obr. 7 Mřížková reprezentace – oblast nad PINCHEm – osazení prvního výměníku

Analogicky se postupuje při osazení sítě dalším výměníkem. Propojí se tedy zbylé dva proudy (Obr. 8), pro které platí, že tepelná kapacita horkého proudu je menší nebo rovna než tepelná kapacita chladného proudu. Opět se dopočítají výkony na jednotlivých proudech. Z Obr. 8 je zřejmé, že výkon druhého výměníku je 1600 kW a je tedy zcela pokryt teplotní interval chladného proudu C1 v oblasti nad PINCHEM. Pomocí výkonů jednotlivých výměníků se dopočítají teploty na horkém proudu H1 a chladném proudu C2.

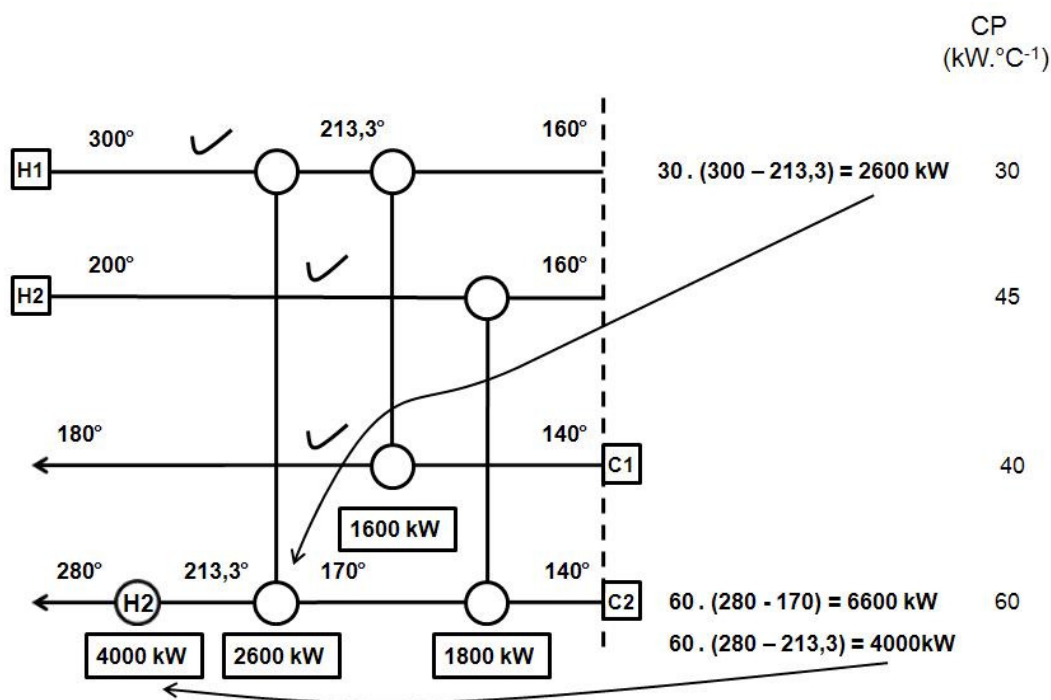


Obr. 8 Mřížková reprezentace – oblast nad PINCHEM – osazení druhého výměníku

Dále se propojí zbylé dva proudy, které ještě nemají zcela pokryté teplotní intervaly, tedy horký proud H1 a chladný proud C2. Opět se dopočítají výkony na jednotlivých proudech. Z Obr. 9 je zřejmé, že výkon třetího výměníku je 2600 kW a je tedy zcela pokryt teplotní interval horkého proudu H1 v oblasti nad PINCHEM.

Opět se pomocí výkonu výměníku dopočítá teplota na chladném proudu C2. Aby byl teplotní interval tohoto proudu zcela pokryt, umístí se zde horký servis H2 o výkonu 4000 kW. Tím je pokryta celá oblast procesní výměny tepla nad PINCHEM.

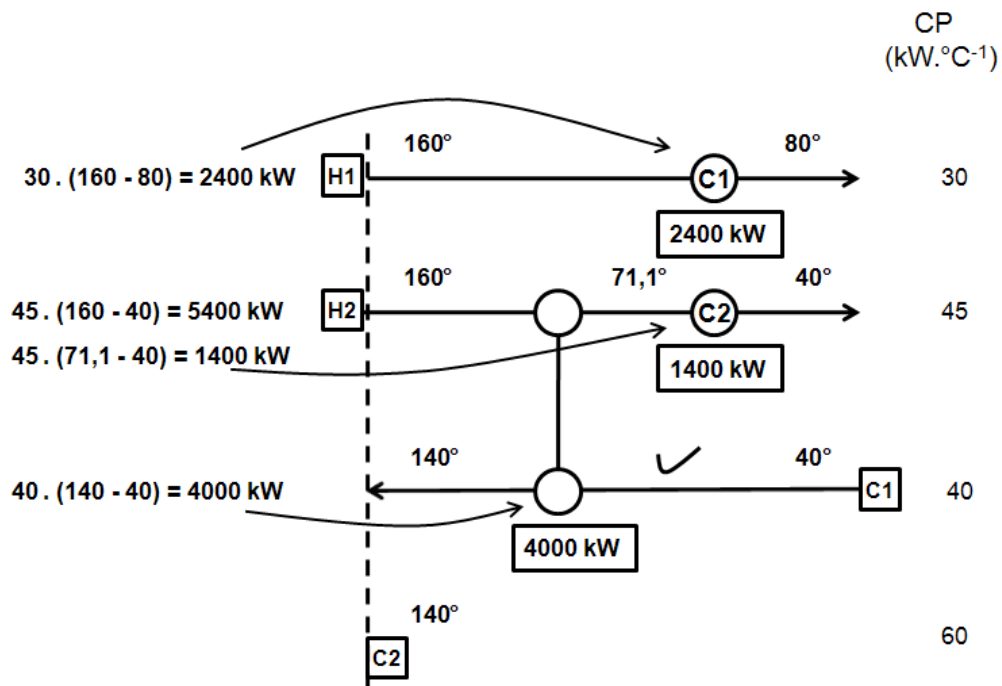
Propojení mezi jednotlivými proudy vždy nemusí představovat jen jeden výměník. V anglické literatuře je toto propojení označováno jako *match* neboli střet, pod kterým si lze představit např. několik sériově zapojených výměníků (*shells*). Pro zjednodušení výkladu v této práci budou však označovány tyto střety zjednodušeně jako výměníky.



Obr. 9 Mřížková reprezentace – oblast nad PINCHem – osazení třetího výměníku

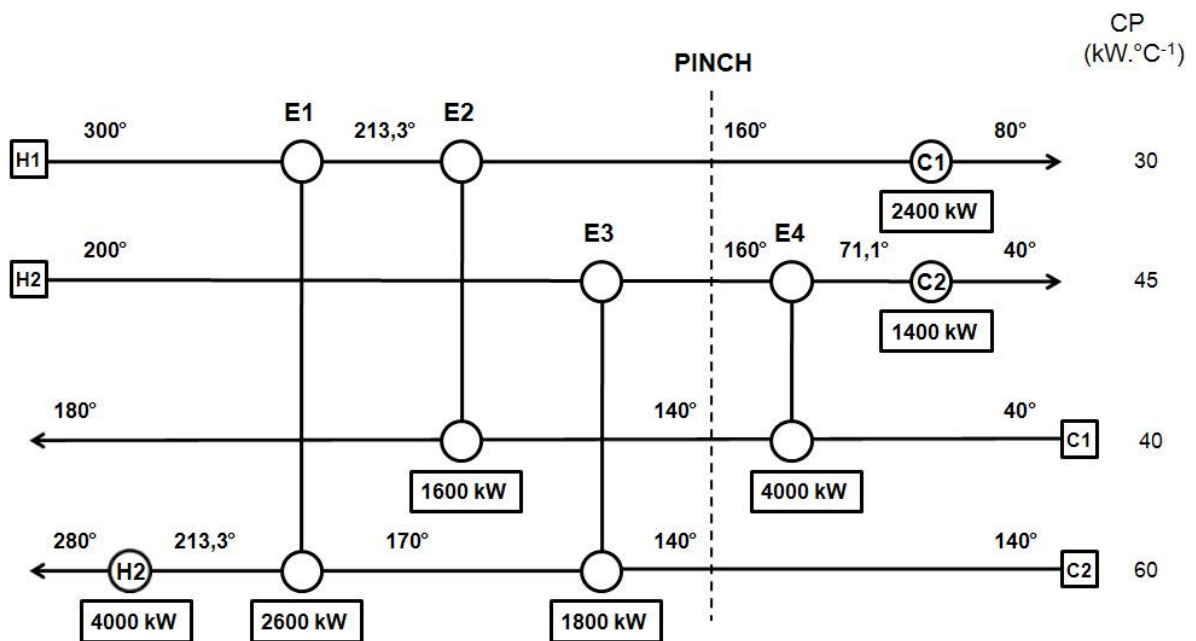
3.2 Řešení oblasti pod PINCHem

Při řešení oblasti pod PINCHem se tedy podle pravidel propojí proudy, kde tepelná kapacita horkého proudu je větší nebo rovna než tepelná kapacita chladného proudu. Pomocí teplotního rozdílu a tepelných kapacit se vypočítá výkon umístěného výměníku. Z Obr. 10 je vidět, že výkon výměníku je 4000 kW a je tedy zcela pokryt teplotní interval chladného proudu C1 v oblasti pod PINCHem. Chladný proud C2 se procesní výměny teple v této oblasti nezúčastňuje. Protože již nelze propojit žádné proudy, umístí se na horké proudy chladný servis, který tyto proudy dochladí na požadované teploty. Tedy na horkém proudu H1 bude chladný servis C1 o výkonu 2400 kW a na horkém proudu H2 bude chladný servis C2 o výkonu 1400 kW.



Obr. 10 Mřížková reprezentace – oblast pod PINCHEm – osazení prvního výměníku

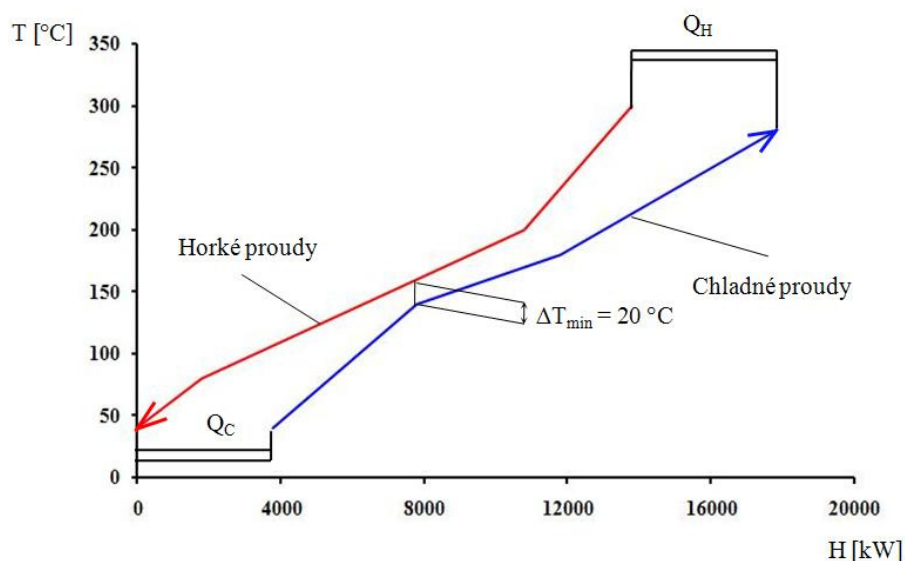
Nyní se tyto dvě řešení, tj. řešení v oblasti nad PINCHEm a v oblasti pod PINCHEm sloučí v jedno výsledné řešení. Toto výsledné řešení sítě výměny tepla pomocí *Pinch Design Method* je na *Obr. 11*.



Obr. 11 Mřížková reprezentace - Výsledné řešení sítě výměny tepla pomocí *Pinch Design Method*

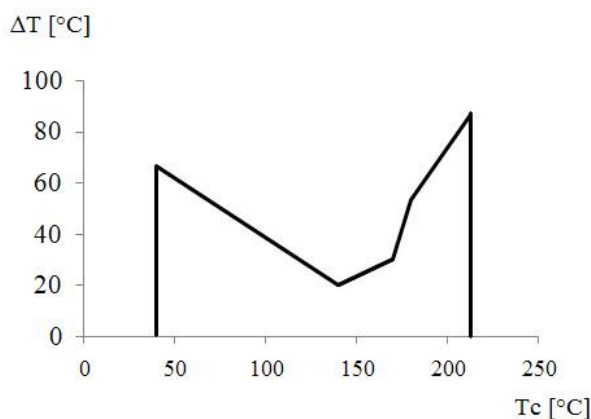
4 Další možnosti *Pinch Design Method*

Návrh sítě výměny tepla pomocí základních pravidel *Pinch Design Method* však nemusí být jednoznačný či optimální. Proto jsou pro tuto metodu vyvinuty další pomocné nástroje, jedním z těchto nástrojů je Diagram hnací síly (*Driving Force Plot - DFP*). Jeho použití bude demonstrováno na příkladu z kapitoly 3 [6]. Kompozitní křivky pro tento příklad jsou znázorněny na Obr. 12.



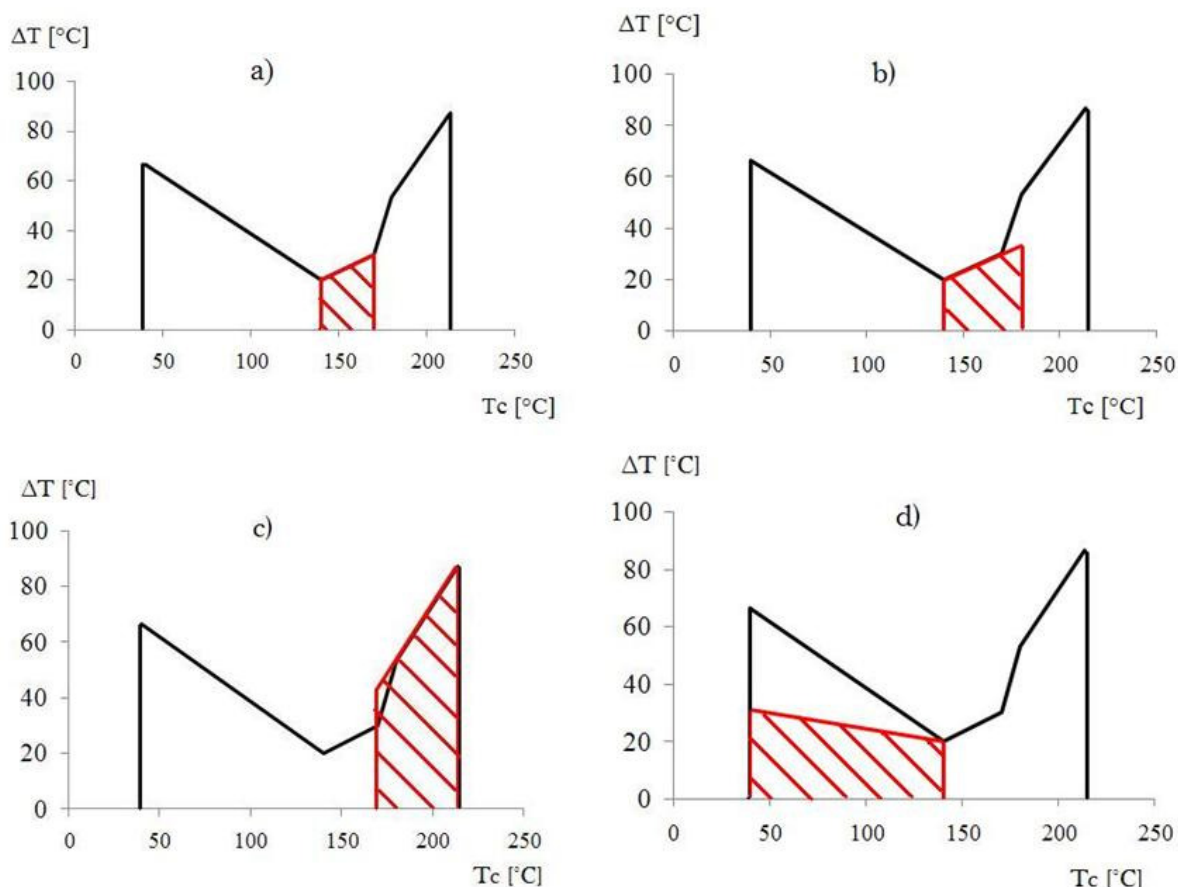
Obr. 12 Kompozitní křivky pro příklad z kapitoly 3

Diagram hnací síly se sestaví z kompozitních křivek. Je to závislost ΔT [°C] na T_C [°C], kde ΔT je rozdíl teplot kompozitních křivek horkých a chladných proudů ve vertikálním směru a T_C je teplota chladných proudů [6]. Diagram hnací síly (Obr. 13) představuje ideální procesní výměnu mezi proudy. S tímto diagramem se budou srovnávat Diagramy hnací síly jednotlivých výměníků a posuzovat jejich ideální umístění v síti.



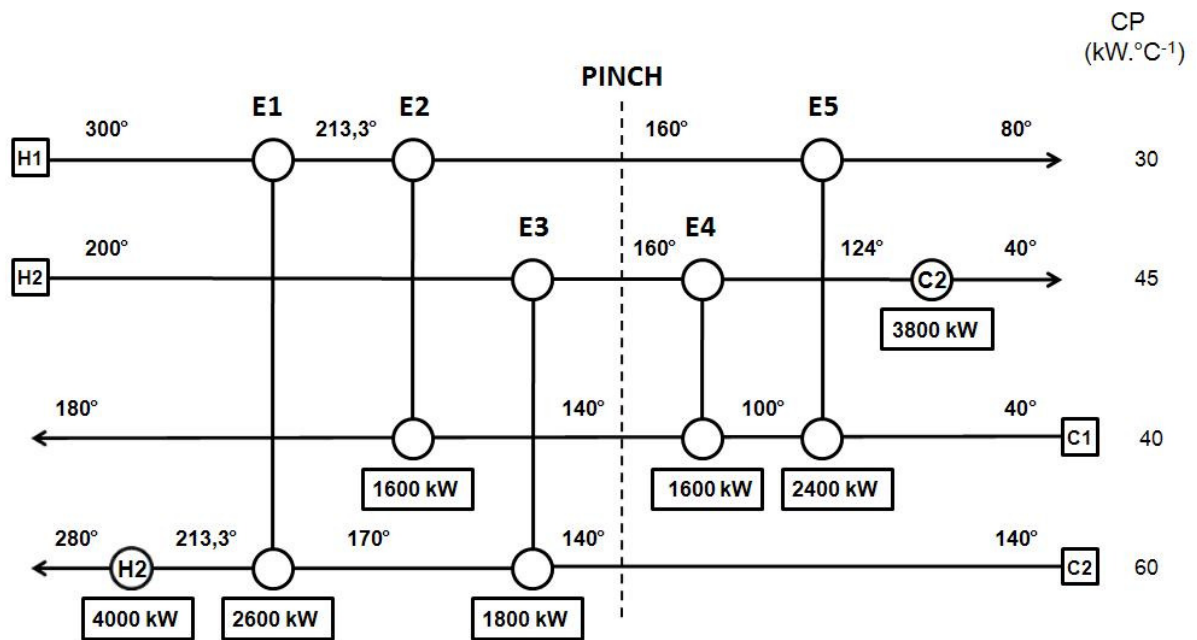
Obr. 13 Diagram hnací síly

Na Obr. 14 jsou znázorněny Diagramy hnací síly pro jednotlivé výměníky (viz Obr. 11 Mřížková reprezentace - Výsledné řešení sítě výměny tepla pomocí Pinch Design Method). Obr. 14a odpovídá výměníku E3, Obr. 14b výměníku E2 a Obr. 14c výměníku E1. U těchto třech Diagramů hnací síly je vidět, že výměníky dobře kopíruje ideální průběh výměny tepla. Zatímco na Obr. 14d, který odpovídá výměníku E4, je vidět, že výměník nepříliš dobře kopíruje ideální průběh.



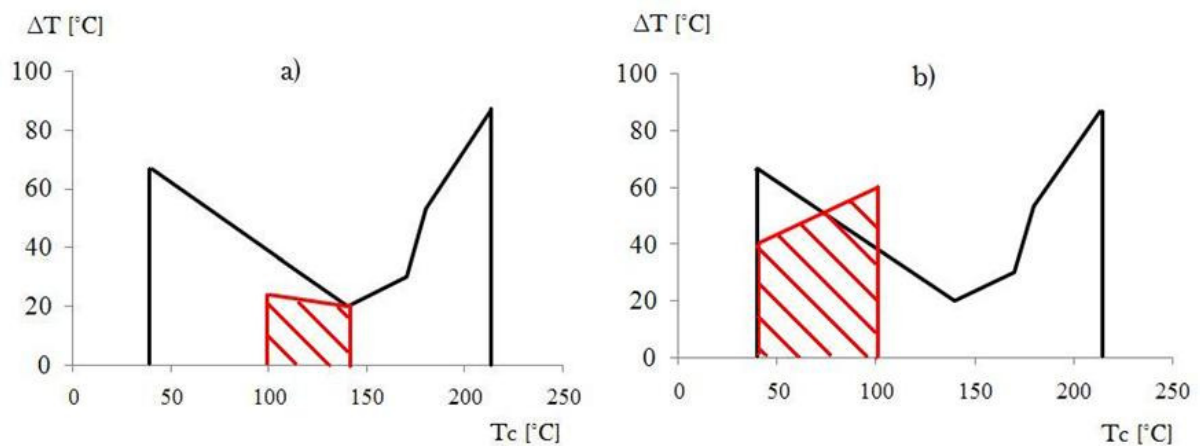
Obr. 14 Diagramy hnací síly – pro čtyři výměníky

Zkusmo se tedy provede změna výkonu výměníku. Např., že se výkon sníží tak, aby zbývající výkon chladného proudu C1 plně pokryl požadavek nekrytého horkého proudu H1, tj. aby se eliminoval chladný servis C1. Mezi chladný proud C1 a horký proud H1 lze tedy umístit výměník E5 o výkonu 2400 kW (Obr. 15). Výkon výměníku E4 se sníží na 1600 kW a výkon chladného servisu C2 se zvýší na 3800 kW.



Obr. 15 Mřížková reprezentace – změna výkonů výměníků

Na Obr. 16 jsou znázorněny Diagramy hnací síly, kde Obr. 16a odpovídá výměníku E4 (1600 kW) a Obr. 16b výměníku E5 (2400 kW). Z těchto Diagramů hnací síly je vidět, že výměníky lépe kopírují ideální průběh výměny tepla.



Obr. 16 Diagramy hnací síly – pro dva výměníky umístěné v oblasti pod PINCHEm

Oba tyto návrhy (čtyři či pět výměníků) jsou shodné co do požadavků na externí energetické zdroje a také v oblasti nad PINCHEm. V oblasti pod PINCHEm se tyto návrhy liší. Který návrh bude v této oblasti vyžadovat menší celkovou plochu výměny tepla, ten bude lepší. Plocha výměny tepla se vypočítá z rovnice výměny tepla [5]:

$$A = \frac{Q}{k \cdot \Delta \bar{T}_{\ln}} \quad , \quad (4)$$

kde A [m^2] je plocha výměny tepla, k [$\text{W}/(\text{m}^2 \cdot \text{K})$] je součinitel prostupu tepla, $\Delta \bar{T}_{\ln}$ [$^{\circ}\text{C}$] je střední logaritmický teplotní rozdíl a Q [W] je celkový přenos tepelného výkonu výměníku.

$$k = \frac{1}{\frac{1}{\alpha_1} + \frac{1}{\alpha_2}} \quad , \quad (5)$$

kde α_1, α_2 [$\text{W}/(\text{m}^2 \cdot \text{K})$] jsou součinitelé přestupu tepla.

$$\Delta \bar{T}_{\ln} = \frac{(T_{11} - T_{22}) - (T_{12} - T_{21})}{\ln \left(\frac{T_{11} - T_{22}}{T_{12} - T_{21}} \right)} \quad , \quad (6)$$

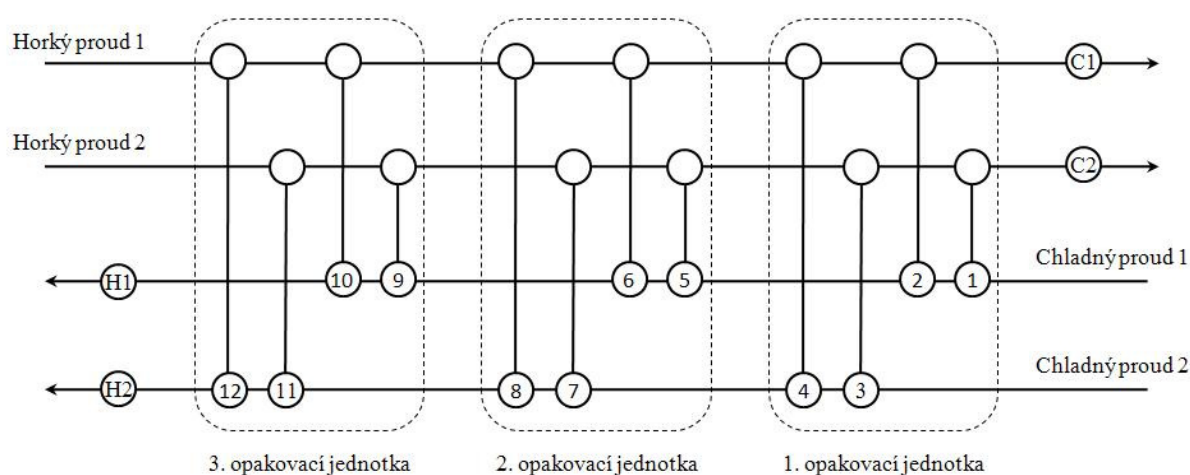
kde T_{11} je vstupní teplota teplejšího proudu, T_{12} je výstupní teplota teplejšího proudu, T_{21} je vstupní teplota chladnějšího proudu a T_{22} je výstupní teplota chladnějšího proudu. Pro tento příklad je α_1 a α_2 voleno $0,2 \text{ kW}/(\text{m}^2 \cdot \text{K})$ [6]. Celková plocha výměny tepla je pak dána součtem ploch výměny tepla jednotlivých výměníků.

Pro návrh řešení pouze metodou PDM vyšla plocha výměny tepla $795,54 \text{ m}^2$, zatímco pro návrh řešení metodou PDM a Diagramem hnacích sil vyšla plocha výměny tepla $607,89 \text{ m}^2$. Úspora plochy je $23,6 \%$. Diagram hnací síly je tedy velmi kvalitním pomocným nástrojem při řešení sítě výměny tepla.

5 Principy deterministické metody

Deterministická metoda je metoda matematického (numerického) programování. Problém syntézy sítě výměny tepla je definován jako systém n_H horkých proudů, které jsou dány horkou vstupní teplotou a musí být ochlazeny na danou výstupní teplotu a n_C chladných proudů, které jsou dány chladnou vstupní teplotou a musí být ohřány na danou výstupní teplotu. Tyto proudy mají také svou konstantní tepelnou kapacitu. Úkolem bude tedy optimalizovat strukturu teplotních výměníků společně s jejich tepelnými výkony, maximalizovat využití tepla mezi horkými a chladnými proudy a následně i minimalizovat výkon externích zdrojů.

Metoda musí obecně popisovat všeobecné charakteristiky, které jsou použitelné na jakýkoliv problém. Aby byly zajištěny veškeré varianty propojení mezi jednotlivými proudy, je v této metodě zaveden pojem „opakovací jednotka sítě výměny tepla“ (*repetitive unit*) [4]. Každá opakovací jednotka se skládá z bloku výměníků tepla zahrnující veškeré možné varianty propojení mezi chladnými a horkými proudy. Obr. 17 ukazuje způsob osazení sítě výměny tepla se dvěma horkými a dvěma chladnými proudy pomocí tří opakovacích jednotek.



Obr. 17 Sít' výměny tepla se třemi opakovacími jednotkami

Každý střet mezi jednotlivými proudy odpovídá výměníku tepla, který bude označen číslem (k), horký proud bude označen číslem (i) a chladný proud číslem (j). Pozici jednotlivého výměníku lze tedy zapsat jako $k(i,j,L)$, kde L je číslo opakovací jednotky.

6 Zpřesněná formulace deterministické metody a její automatizace

Pro řešení sítě výměny tepla pomocí automatizovaného modelu je důležité popsat systém pomocí účelové funkce a omezujících podmínek. Optimalizační problém sítě výměny tepla zahrnuje n_C+n_H omezení ve formě rovnosti, čtyři omezení ve formě nerovnosti a dvě logická omezení pro každý výměník k v síti.

6.1 Účelová funkce

Při řešení sítě výměny tepla je zpravidla účelová funkce Z formulována ve smyslu procesní výměny tepla (výkon výměňkové sítě). Aby bylo dosaženo maximální procesní výměna tepla (*Maximum Energy Recovery - MER*), hledá se maximum této účelové funkce. Účelová funkce je tedy ve tvaru [4]:

$$\max Z = \sum_{k=1}^{n_E} X_k, \quad (7)$$

kde X_k [W] je výkon výměníku tepla k a n_E je počet výměníků v síti.

6.2 Omezení ve formě rovnosti

Toto omezení vychází z rovnice tepelné bilance. Celkový tepelný tok, který je potřebný k ohřátí chladného proudu j ze vstupní teploty na teplotu výstupní musí odpovídat součtu výkonů výměníků tepel a horkého servisu. Je dáno rovnicí (8) [4]:

$$CP_{C(j)} \cdot (T_{C(j)}^t - T_{C(j)}^s) - \sum_{\substack{k=1 \\ c(k)=j}}^{n_E} X_k - H_j = 0 \quad (j = 1, \dots, n_C), \quad (8)$$

kde $T_{C(j)}^s$ [°C], $T_{C(j)}^t$ [°C] je vstupní, resp. výstupní teplota chladného proudu j , $CP_{C(j)}$ [W/°C] je tepelná kapacita chladného proudu j , X_k [W] je výkon výměníku k , H_j [W] je výkon horkého servisu a index c_k značí chladný proud tekoucí skrz výměník k .

Podobně, celkový tepelný tok, který je potřebný k ochlazení horkého proudu i ze vstupní teploty na teplotu výstupní musí odpovídat součtu výkonů výměníků tepel a chladných servisů. Je dáno rovnicí (9) [4]:

$$CP_{H(i)} \cdot (T_{H(i)}^s - T_{H(i)}^t) - \sum_{\substack{k=1 \\ h(k)=i}}^{n_E} X_k - C_i = 0 \quad (i = 1, \dots, n_H), \quad (9)$$

kde $T_{H(i)}^s$ [°C], $T_{H(i)}'$ [°C] je vstupní, resp. výstupní teplota horkého proudu i , $CP_{H(i)}$ [W/°C] je tepelná kapacita horkého proudu i , X_k [W] je výkon výměníku k , C_i [W] je výkon chladného servisu a index h_k značí horký proud tekoucí skrz výměník k .

6.3 Logická omezení

Logická omezení využívají binárních proměnných, které určují existenci každého výměníku v síti. Logická omezení jsou ve tvaru [4]:

$$G(1-\delta_k) - X_k \geq 0 \quad (k=1, \dots, n_E), \quad (10)$$

$$X_k + \varepsilon(\delta_k - 1) \geq 0 \quad (k=1, \dots, n_E), \quad (11)$$

kde G [W], ε [W] je maximální, resp. minimální dovolený výkon jednotlivých výměníků v síti. δ_k je binární proměnná označující přítomnost daného výměníku. Jestliže $\delta_k = 1$ výměník neexistuje, je-li $\delta_k = 0$ výměník existuje a bude se podílet na výměně tepla.

6.4 Omezení ve formě nerovnosti

Pro protiproudý výměník tepla vycházíme z rovnic teplotních rozdílů na koncích výměníků pro dodržení zadané hodnoty ΔT_{\min} [4].

- Pro horký konec výměníku platí:

$$T_{hot_{in}} - T_{cold_{out}} \geq \Delta T_{\min}, \quad (12)$$

- Pro chladný konec výměníku platí:

$$T_{hot_{out}} - T_{cold_{in}} \geq \Delta T_{\min}, \quad (13)$$

kde $T_{hot_{in}}$ [°C], $T_{hot_{out}}$ [°C] je vstupní, resp. výstupní teplota na horkém konci výměníku a $T_{cold_{in}}$ [°C], $T_{cold_{out}}$ [°C] je vstupní, resp. výstupní teplota na chladném konci výměníku.

Pro tento matematický model se předchozí rovnice (12), (13) zapíšou ve tvaru [4]:

- Pro horký konec výměníku platí:

$$T_{hot_{in}} - T_{cold_{out}} \geq \delta_k \cdot M + \Delta T_{\min}, \quad (14)$$

- Pro chladný konec výměníku platí:

$$T_{hot_{out}} - T_{cold_{in}} \geq \delta_k \cdot M + \Delta T_{min}, \quad (15)$$

kde M [°C] je manipulační konstanta teplotního rozdílu. Je to dostatečně malé záporné číslo k anulování omezení ve formě nerovnosti.

Po rozepsání rovnic (14) a (15) se tedy získá omezení ve formě nerovnosti [4]:

- Pro horký konec výměníku platí:

$$\frac{1}{CP_{H,h(k)}} \cdot \left(C_{h(k)} + \sum_{\substack{l=1 \\ h(l)=h(k)}}^k X_l \right) - \frac{1}{CP_{C,c(k)}} \cdot \sum_{\substack{l=1 \\ c(l)=c(k)}}^k X_l - \Delta T_{min} - \delta_k \cdot M + T_{H,h(k)}^t - T_{C,c(k)}^t \geq 0 \quad (16)$$

$(k = 1, \dots, n_E)$

- Pro chladný konec výměníku platí:

$$\frac{1}{CP_{H,h(k)}} \cdot \left(C_{h(k)} + \sum_{\substack{l=1 \\ h(l)=h(k)}}^{k-1} X_l \right) - \frac{1}{CP_{C,c(k)}} \cdot \sum_{\substack{l=1 \\ c(l)=c(k)}}^{k-1} X_l - \Delta T_{min} - \delta_k \cdot M + T_{H,h(k)}^t - T_{C,c(k)}^t \geq 0 \quad (17)$$

$(k = 1, \dots, n_E)$

Mezi další omezení ve formě nerovnosti patří nastavení maximálního počtu výměníků tepla v síti (rovnice (18)) a omezení kladného výkonu výměníků (rovnice (19)) [4]:

$$N_{max} - \sum_{k=1}^{n_E} (1 - \delta_k) \geq 0, \quad (18)$$

$$X_k \geq 0 \quad (k = 1, \dots, n_E), \quad (19)$$

kde N_{max} [-] je nastavitelný maximální počet výměníků tepla v síti.

6.5 Ilustrativní ukázka konkrétní formulace jednotlivých omezení

Pro názornost se tyto tři typy omezujících podmínek objasní na výměňkové síti z Obr. 17. Omezující podmínku typu rovnosti, tj. rovnice (8), lze pro *horký proud 1* zapsat následovně:

$$CP_{H(1)} \cdot (T_{H(1)}^s - T_{H(1)}^t) - (X_{12} + X_{10} + X_8 + X_6 + X_4 + X_2 + C_1) = 0 \quad (20)$$

Logická omezení, tj. rovnice (10), (11), lze např. pro výměník 8 zapsat následovně:

$$G(1 - \delta_8) - X_8 \geq 0 \quad (21)$$

$$X_8 + \varepsilon(\delta_8 - 1) \geq 0 \quad (22)$$

Omezující podmínku typu nerovnosti, tj. rovnice (16), (17) lze např. pro výměník 8 zapsat následovně:

- Pro horký konec výměníku 8 platí:

$$\begin{aligned} & \frac{1}{CP_{H(1)}} \cdot (C_1 + X_2 + X_4 + X_6 + X_8) - \frac{1}{CP_{C(2)}} \cdot (X_3 + X_4 + X_7 + X_8) - \\ & -\Delta T_{\min} - \delta_8 \cdot M + T_{H(1)}^t - T_{C(2)}^s \geq 0 \end{aligned} \quad (23)$$

- Pro chladný konec výměníku 8 platí:

$$\begin{aligned} & \frac{1}{CP_{H(1)}} \cdot (C_1 + X_2 + X_4 + X_6) - \frac{1}{CP_{C(2)}} \cdot (X_3 + X_4 + X_7) - \\ & -\Delta T_{\min} - \delta_8 \cdot M + T_{H(1)}^t - T_{C(2)}^s \geq 0 \end{aligned} \quad (24)$$

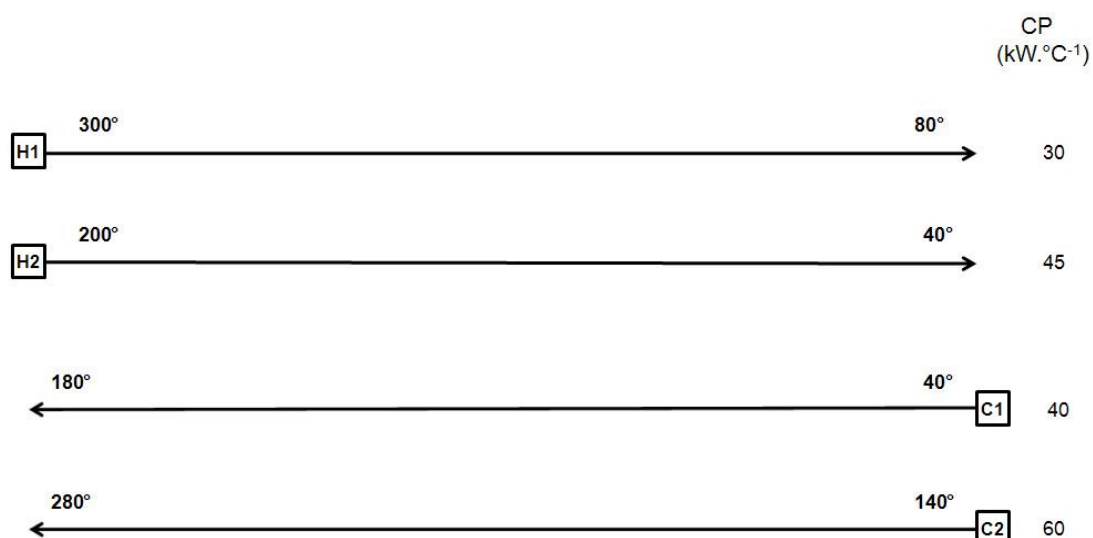
Na základě rovnic uvedených v předchozích kapitolách byl vytvořen automatizovaný model pro řešení syntézy výměny tepla pomocí softwaru *Maple*, který bude testován na několika následujících příkladech.

Pro přehlednější práci se softwarem *Maple* byl vytvořen soubor (HENExprGen.mpl) sdružující vytvořené procedury pro automatizované generování rovnic (viz Příloha 1). Řešení každé konkrétní úlohy tak lze jednoduše a velmi přehledně provádět v mapleovském *worksheetu* (*worksheets* pro konkrétní příklady viz Přílohy 2 až 5), který tento soubor pro automatizované generování rovnic volá.

7 Ukázka aplikace deterministické metody

V této kapitole se bude testovat stejný příklad, který byl použit pro ukázkou aplikace *Pinch Design Method* (kapitola 3), tj. příklad z článku [6] (*Case 1*).

Při aplikaci deterministického modelu vyjdeme ze zadání tohoto příkladu formulovaného v kapitole 3 v *Tab. 1*. Tomuto zadání odpovídá následující mřížková reprezentace (*Obr. 18*).



Obr. 18 Mřížková reprezentace pro aplikaci deterministického modelu

Pro použití navrhovaného modelu v této práci je důležitý zvolený počet opakovacích jednotek (viz kapitola 5). Ve vytvořeném automatizovaném modelu se postupně manuálně zvyšuje počet opakovacích jednotek z jedné na tři (viz Příloha 2 pro 1 opakovací jednotku). Z *Tab. 2* je vidět, že pro jednu, dvě i tři opakovací jednotky vychází maximální procesní výměny tepla (MER) i výkony externích energetických zdrojů (Q_C , Q_H) konstantní, proto postačí použít jen jednu opakovací jednotku.

Počet opakovacích jednotek	MER [kW]	Q_C [kW]	Q_H [kW]
1	10000	3800	4000
2	10000	3800	4000
3	10000	3800	4000

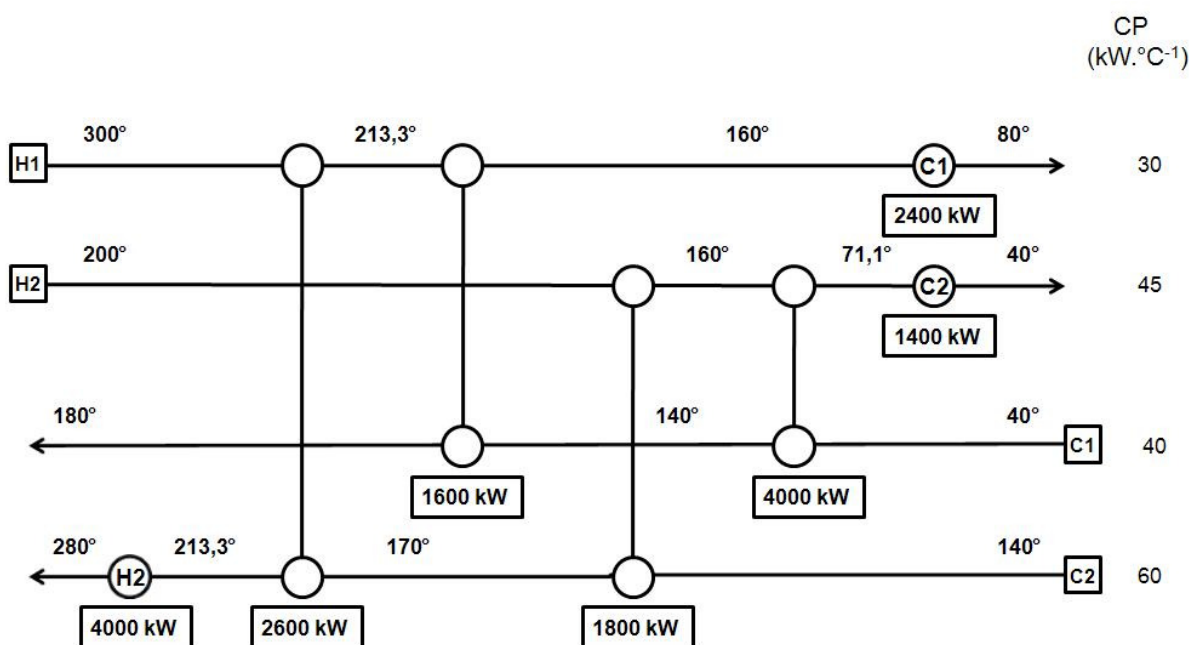
Tab. 2 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu opakovacích jednotek

Postupně se (vhodným nastavením parametru N_{max} – viz rovnice (19) v kapitole 6.4) manuálně snižuje maximální dovolený počet použitých výměníků n_E v síti ze čtyř až na jeden (viz například Příloha 2 pro 4 výměníky v síti). Mění se dovolený počet výměníků v síti ovlivňuje výkon výměňkové sítě (MER). Jak je vidět z *Tab. 3*, maximální procesní výměny tepla (MER) se dosáhne v tomto případě aplikováním jedné opakovací jednotky při použití čtyř výměníků (n_E), dvou chladných externích energetických zdrojů (n_C) a jednoho horkého externího energetického zdroje (n_H).

Počet opakovacích jednotek	n_E [ks]	n_C [ks]	n_H [ks]	MER [MW]	Q_C [MW]	Q_H [MW]
1	4	2	1	10000	3800	4000
1	3	2	2	9100	4700	4900
1	2	2	2	8000	5800	6000
1	1	2	2	4000	9800	10000

Tab. 3 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků

Z *Obr. 19*, který znázorňuje výsledky dosažené deterministickým modelem, je vidět, že osazení výměňkové sítě, výkony jednotlivých výměníků i externích energetických zdrojů jsou stejné jako při použití *Pinch Design Method* (viz kapitola 3, *Obr. 11*).



Obr. 19 Mřížková reprezentace - výsledné řešení sítě výměny tepla obdržené deterministickým modelem.

8 Řešení praktických příkladů

V této kapitole se budou pomocí programové implementace sestaveného deterministického modelu řešit postupně tři praktické příklady.

8.1 Příklad 1

Zadání příkladu 1, publikovaného v [7] je v *Tab. 4*.

proud	CP [MW/°C]	T _{in} [°C]	T _{out} [°C]
H1	0,15	250	40
H2	0,25	200	80
C1	0,2	20	180
C2	0,3	140	230
$\Delta T_{\min} = 10 \text{ }^{\circ}\text{C}$			

Tab. 4 Vstupní hodnoty pro příklad 1

V automatizovaném modelu (viz Příloha 3 pro 2 opakovací jednotky) se postupně zvyšuje počet opakovacích jednotek z jedné na tři. Z *Tab. 5* je vidět, že maximální procesní výměny tepla (MER) i minimálních výkonů externích energetických zdrojů (Q_C , Q_H) se dosáhne již při použití dvou opakovacích jednotek, neboť při následném zvyšování počtu opakovacích jednotek se tyto hodnoty již nemění.

Počet opakovacích jednotek	MER [MW]	Q_C [MW]	Q_H [MW]
1	49,875	11,625	9,125
2	51,5	10	7,5
3	51,5	10	7,5

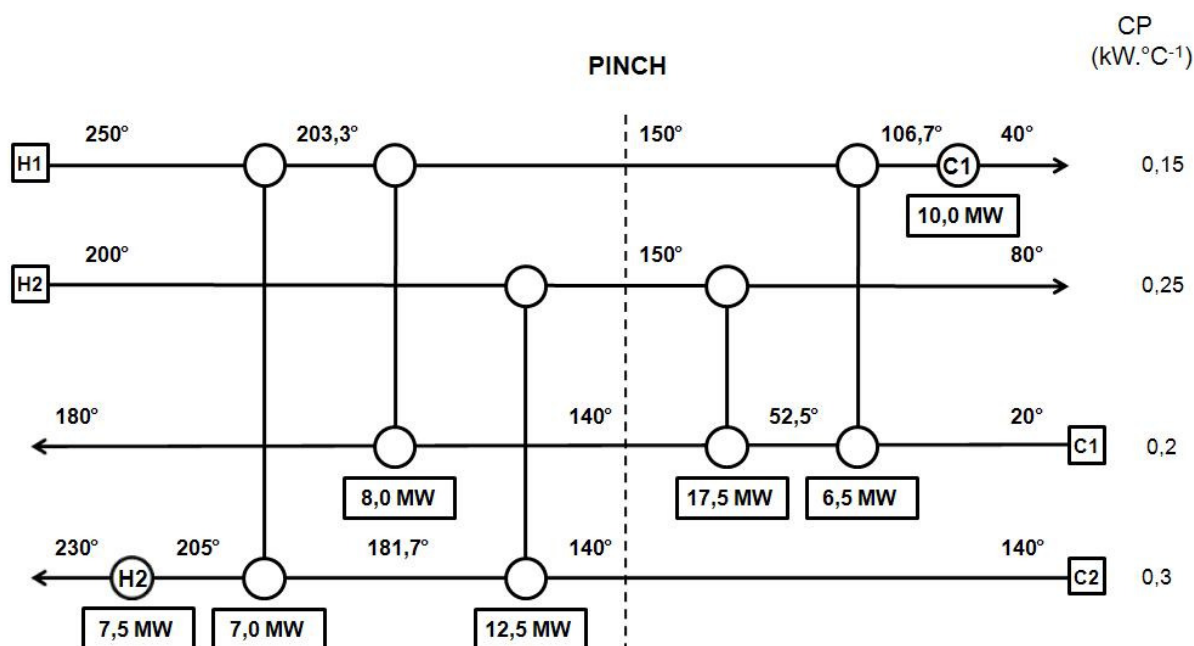
Tab. 5 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu opakovacích jednotek pro příklad 1

Pro nalezené dvě opakovací jednotky se postupně snižuje maximální dovolený počet použitých výměníků v síti z osmi až na čtyři (viz Příloha 3 pro 5 výměníků v síti). Jak je vidět z *Tab. 6*, pro dosažení maximální procesní výměny tepla (MER) postačuje použití pěti výměníků, jednoho chladného externího energetického zdroje a jednoho horkého externího energetického zdroje.

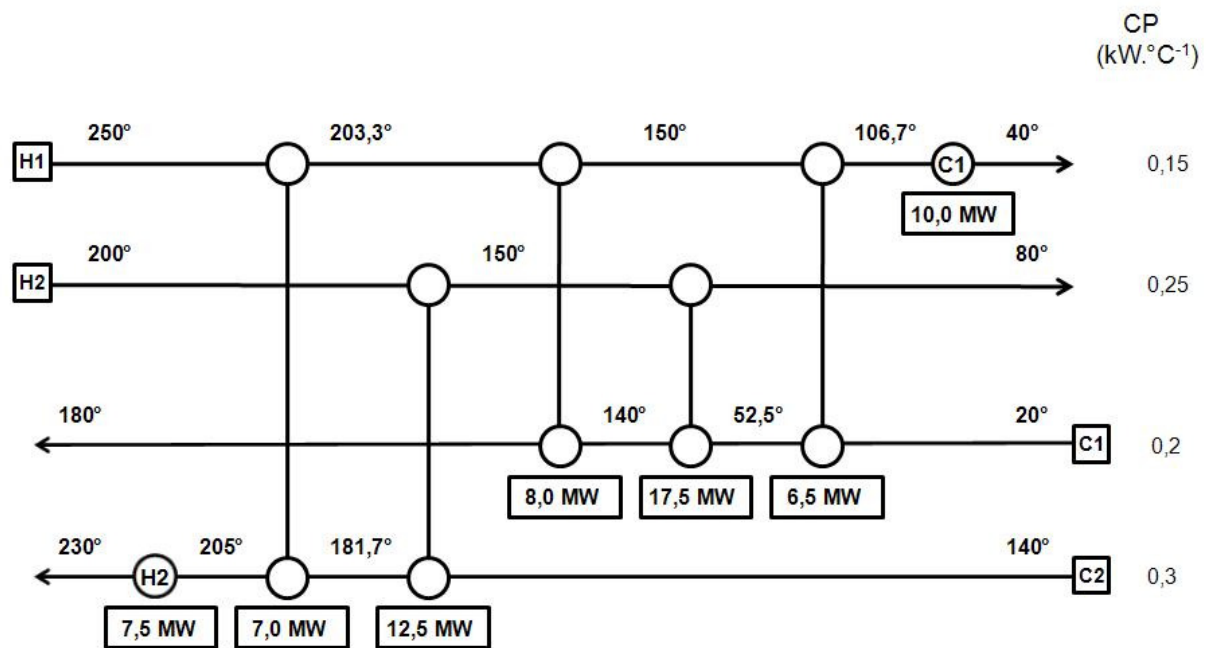
Počet opakovacích jednotek	n_E [ks]	n_C [ks]	n_H [ks]	MER [MW]	Q_C [MW]	Q_H [MW]
2	8	1	1	51,5	10	7,5
2	7	1	1	51,5	10	7,5
2	6	1	1	51,5	10	7,5
2	5	1	1	51,5	10	7,5
2	4	1	2	49,875	11,625	9,125

Tab. 6 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků pro příklad 1

Na Obr. 20 je mřížková reprezentace sítě výměny tepla při aplikaci *Pinch Design Method* [7] a na Obr. 21 při aplikaci deterministického modelu. Ze srovnání těchto obrázků (Obr. 20, Obr. 21) je zřejmé, že oběma metodami lze dosáhnout stejného osazení sítě jednotlivými výměníky tepla a totožných výkonů těchto výměníků a výkonů externích energetických zdrojů.



Obr. 20 Mřížková reprezentace při aplikaci *Pinch Design Method* [7] pro příklad 1



Obr. 21 Mřížková reprezentace při aplikaci deterministického modelu pro příklad 1

8.2 Příklad 2

Vstupní hodnoty pro příklad 2 byly převzaty z literatury [4] (kde je tento příklad označen jako tzv. *Case Study 1*) a jsou uvedeny v Tab. 7.

proud	CP [MW/°C]	T_{in} [°C]	T_{out} [°C]
H1	2	145	60
H2	4	140	65
C1	3	40	120
C2	5	80	100
C3	1,5	40	100
$\Delta T_{min} = 20$ °C			

Tab. 7 Vstupní hodnoty k příkladu 2 dle [4] (*Case study 1*)

Při řešení tohoto příkladu se postupuje podobně jako u předchozího příkladu. V automatizovaném modelu (viz Příloha 4 pro 3 opakovací jednotky) se postupně zvyšuje počet opakovacích jednotek z jedné na čtyři. Z Tab. 8 je vidět, že maximální procesní výměny tepla (MER) i minimálních výkonů externích energetických zdrojů (Q_C , Q_H) se dosáhne při použití tří opakovacích jednotek, neboť při následném zvyšování počtu opakovacích jednotek se tyto hodnoty nemění.

Počet opakovacích jednotek	MER [kW]	Q_C [kW]	Q_H [kW]
1	370	100	60
2	426	44	4
3	430	40	0
4	430	40	0

Tab. 8 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu opakovacích jednotek k příkladu 2

Postupně se snižuje maximální dovolený počet použitých výměníků (viz Příloha 4 pro 6 výměníků v síti). Jak je vidět z Tab. 9, maximální procesní výměny tepla (MER) se dosáhne již při použití šesti výměníků, dvou chladných externího energetického zdroje a žádného horkého externího energetického zdroje.

Počet opakovacích jednotek	n_E [ks]	n_C [ks]	n_H [ks]	MER [MW]	Q_C [MW]	Q_H [MW]
3	9	2	0	430	40	0
3	8	2	0	430	40	0
3	7	2	0	430	40	0
3	6	2	0	430	40	0
3	5	2	1	422,5	47,5	7,5
3	4	2	2	390	80	40
3	3	2	2	330	140	100

Tab. 9 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků k příkladu 2 při aplikaci deterministického modelu

Tab. 10 ukazuje pro srovnání závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků, jak byla pro tento příklad prezentována v literatuře [4]. K řešení tohoto příkladu použili autoři [4] software *LINDO*. Při srovnání dosažených výsledků z obou tabulek je vidět, že hodnoty se od sebe liší. Jedná se však o hodnoty, které jsou až pod optimálním řešením, tj. od pěti použitých výměníků níže. Při podrobnějším testování se zjistilo, že tyto hodnoty závisí na zvoleném maximálním dovoleném výkonu jednotlivých výměníků (parametr G [W] - viz kapitola 6.3). Např. pokud se G zvolí 130 kW, vyjde pro čtyři výměníky MER 390 kW (viz Tab. 9). Pokud se G zvolí 200 kW, vyjde pro čtyři výměníky MER 407,5 kW. Protože v literatuře [4] autoři

neuvědli, jaké nastavení parametru G použili, hodnoty v obou tabulkách (Tab. 9, Tab. 10) se liší.

Počet opakovacích jednotek	n_E [ks]	n_C [ks]	n_H [ks]	MER [MW]	Q_C [MW]	Q_H [MW]
3	9	2	0	430	40	0
3	8	2	0	430	40	0
3	7	2	0	430	40	0
3	6	2	0	430	40	0
3	5	2	1	423	47	7
3	4	2	2	409	61	21

Tab. 10 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků k příkladu 2 dle literatury [4] (Case study 1)

8.3 Příklad 3

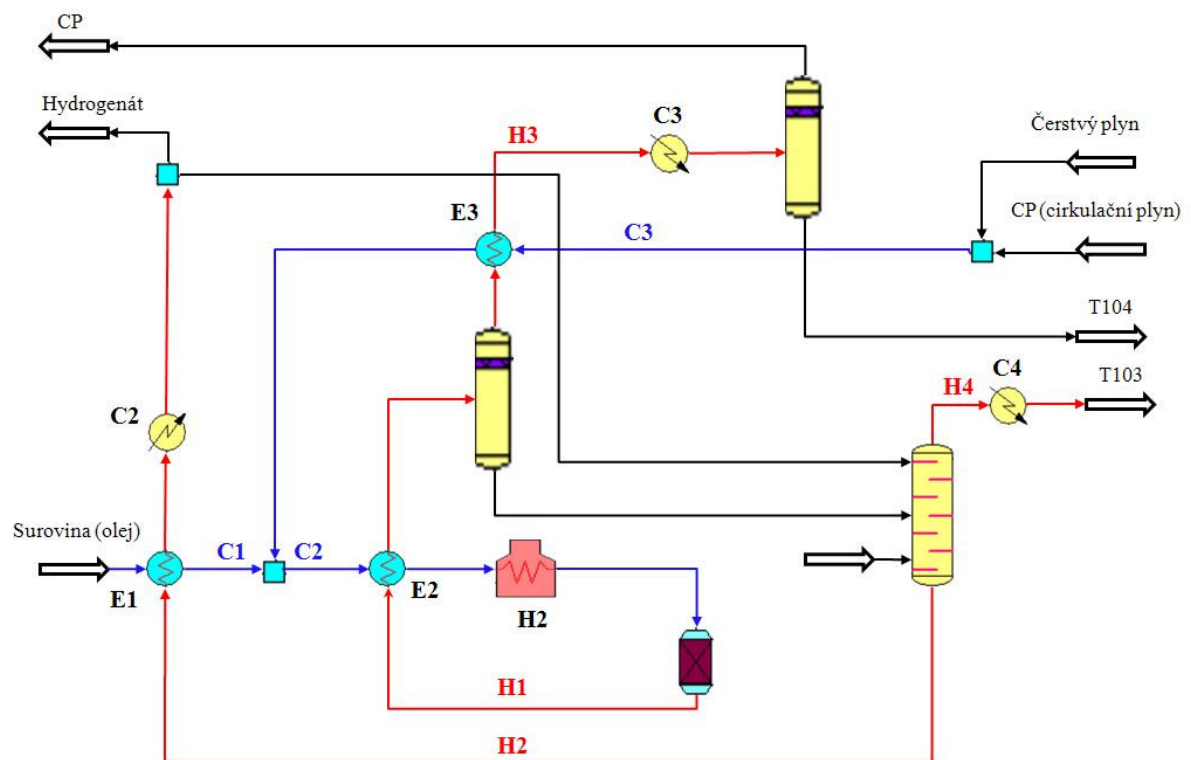
Zadání tohoto příkladu je z literatury [1]. Jedná se o proces hydrogenační rafinace oleje. Z celkového počtu 23 procesních proudů bylo v literatuře [1] vybráno 7, které jsou významné z hlediska výměny tepla. Zjednodušené technologické schéma je na Obr. 22. Autor práce [1] řeší pro toto technologické schéma rekonstrukci stávající výměňkové sítě (Obr. 23) pro nové provozní požadavky.

Stávající uspořádání výměňkové sítě (Obr. 23) bylo navrženo projekční organizací svou vlastní blíže nespecifikovanou metodou. Účelem tohoto příkladu je provést alternativní návrh výměňkové sítě tohoto procesu pomocí vytvořeného deterministického modelu a porovnat obdržené řešení s originálním projekčním návrhem.

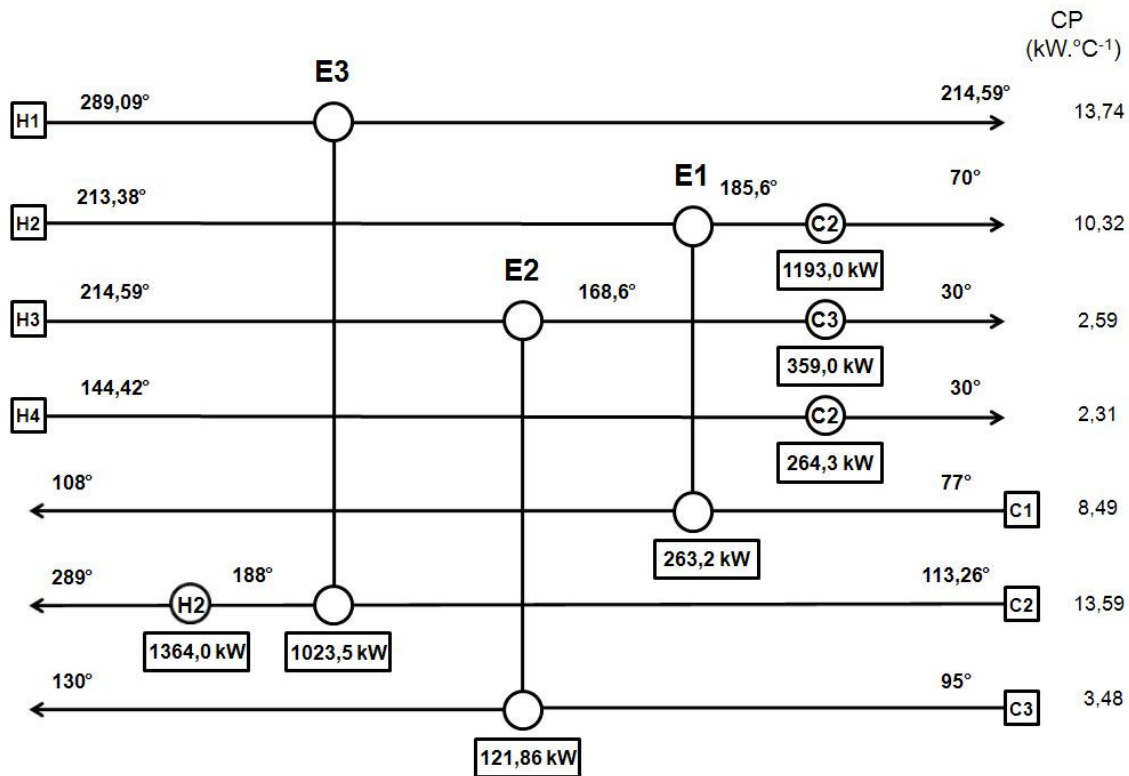
Vstupní data pro tento příklad jsou převzaty z [1] a jsou uvedeny v Tab. 11. Pro nastavení teplotního přiblížení pro srovnatelný návrh deterministickým modelem je z Obr. 23 identifikován minimální teplotní rozdíl mezi horkými a chladnými proudy výměňkové sítě. Nachází se na chladné straně výměníku E2, jeho hodnota činí 73,6 °C (168,6 – 95,0 = 73,6). Proto je tato hodnota v příkladu brána jako dovolené minimální teplotní přiblížení výměníků tepla umístěných v síti pro aplikaci deterministického modelu.

proud	CP [kW/°C]	T _{in} [°C]	T _{out} [°C]
H1	13,74	289,09	214,59
H2	10,32	213,38	70,00
H3	2,59	214,59	30,00
H4	2,31	144,42	30,00
C1	8,49	77,00	108,00
C2	13,59	113,26	289,00
C3	3,48	95,00	130,00

Tab. 11 Vstupní hodnoty k příkladu 3 [1] (Hydrogenační rafinace oleje)



Obr. 22 Technologické schéma hydrogenační rafinace oleje [1]



Obr. 23 Mřížková reprezentace - hydrogenační rafinace oleje – původní projekční návrh [1]

Při řešení tohoto příkladu se postupuje podobně jako u předchozích příkladů. V automatizovaném modelu (viz Příloha 5 pro 1 opakovací jednotku) se postupně zvyšuje počet opakovacích jednotek z jedné na dvě. Z Tab. 12 je vidět, že pro jednu i dvě opakovací jednotky vychází maximální procesní výměna tepla (MER) i minimální výkony externích energetických zdrojů (Q_C , Q_H) konstantní, proto postačí pro další řešení použít jen jednu opakovací jednotku.

Počet opakovacích jednotek	MER [kW]	Q_C [kW]	Q_H [kW]
1	1679,6	1566,1	1093,7
2	1679,6	1566,1	1093,7

Tab. 12 Závislost maximální procesní výměny tepla (MER) a výkonů externích energetických zdrojů (Q_C , Q_H) na počtu opakovacích jednotek k příkladu 3

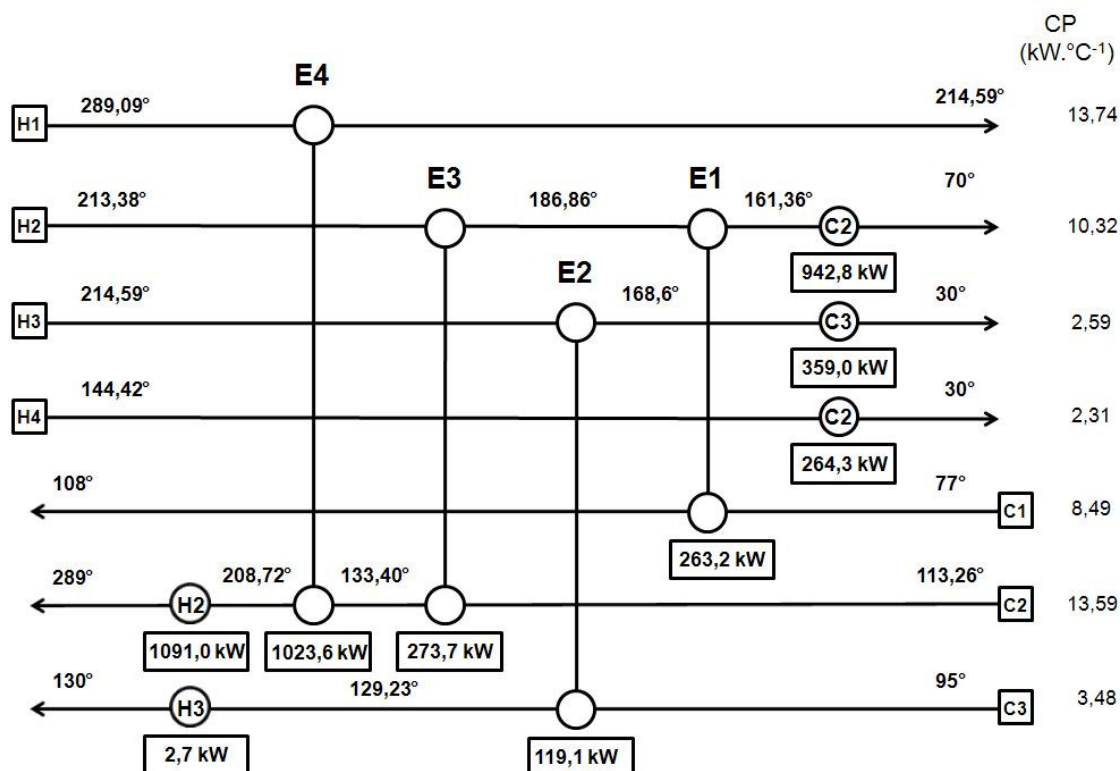
V rámci jedné opakovací jednotky se pak v deterministickém modelu postupně snižuje maximální dovolený počet použitých výměníků (viz Příloha 5 pro 4 výměníky v síti). Jak je vidět z Tab. 13, maximální procesní výměny tepla (MER) se dosáhne již při použití čtyř

výměníků, tří chladných externích energetických zdrojů a dvou horkých externích energetických zdrojů.

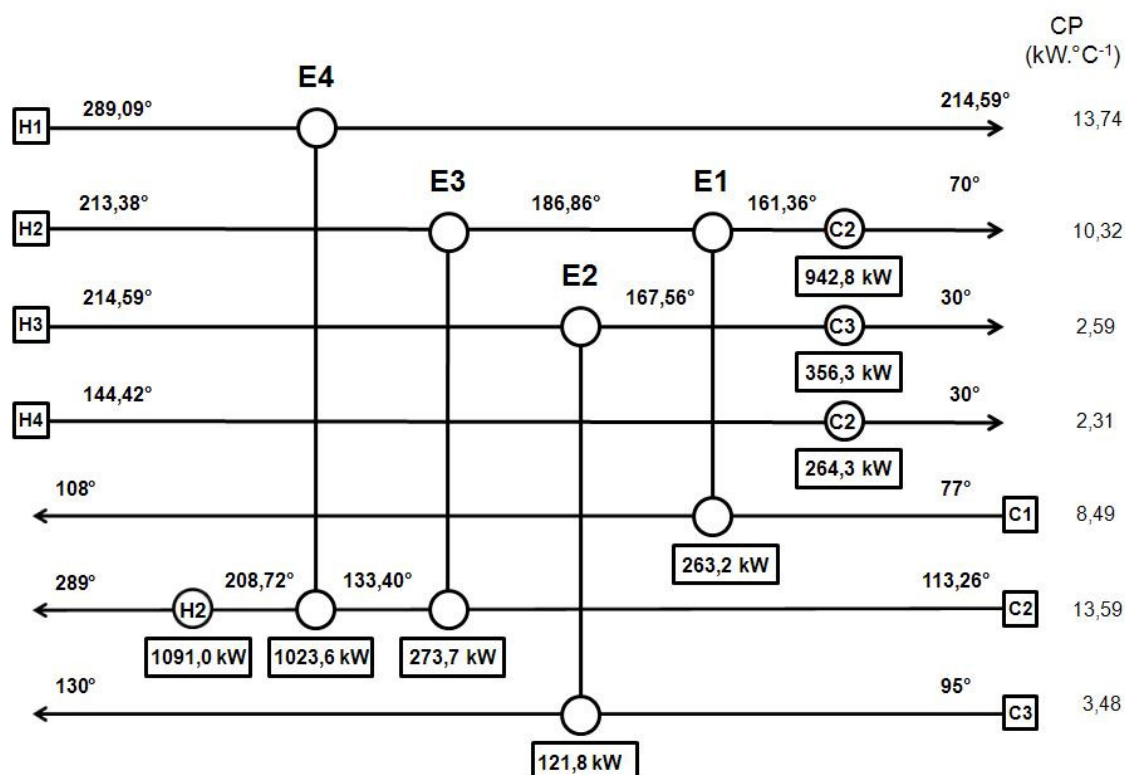
Počet opakovacích jednotek	n_E [ks]	n_C [ks]	n_H [ks]	MER [MW]	Q_C [MW]	Q_H [MW]
1	5	3	2	1679,6	1566,1	1093,7
1	4	3	2	1679,6	1566,1	1093,7
1	3	3	2	1560,5	1685,2	1212,8
1	2	3	3	1297,3	1948,4	1476,0
1	1	3	3	1023,6	2222,1	1749,7

Tab. 13 Závislost maximální procesní výměny tepla (MER) a počtu externích energetických zdrojů (Q_C , Q_H) na počtu použitých výměníků k příkladu 3

Mřížková reprezentace sítě výměny tepla získaná deterministickým modelem je na Obr. 24. Na chladném proudu C3 vyšel horký servis H3 o výkonu 2,7 kW (ohřátí proudu o 0,77 °C). Tato hodnota je příliš malá na to, aby mohla být prakticky realizována. Pro praktickou aplikaci obdržených výsledků je proto vhodné zvýšit výkon výměníku E2 na 121,8 kW, tím se eliminuje horký servis H3 a sníží výkon chladného servisu C3 na 359,0 kW (Obr. 25).



Obr. 24 Mřížková reprezentace hydrogenační rafinace oleje při aplikaci deterministického modelu

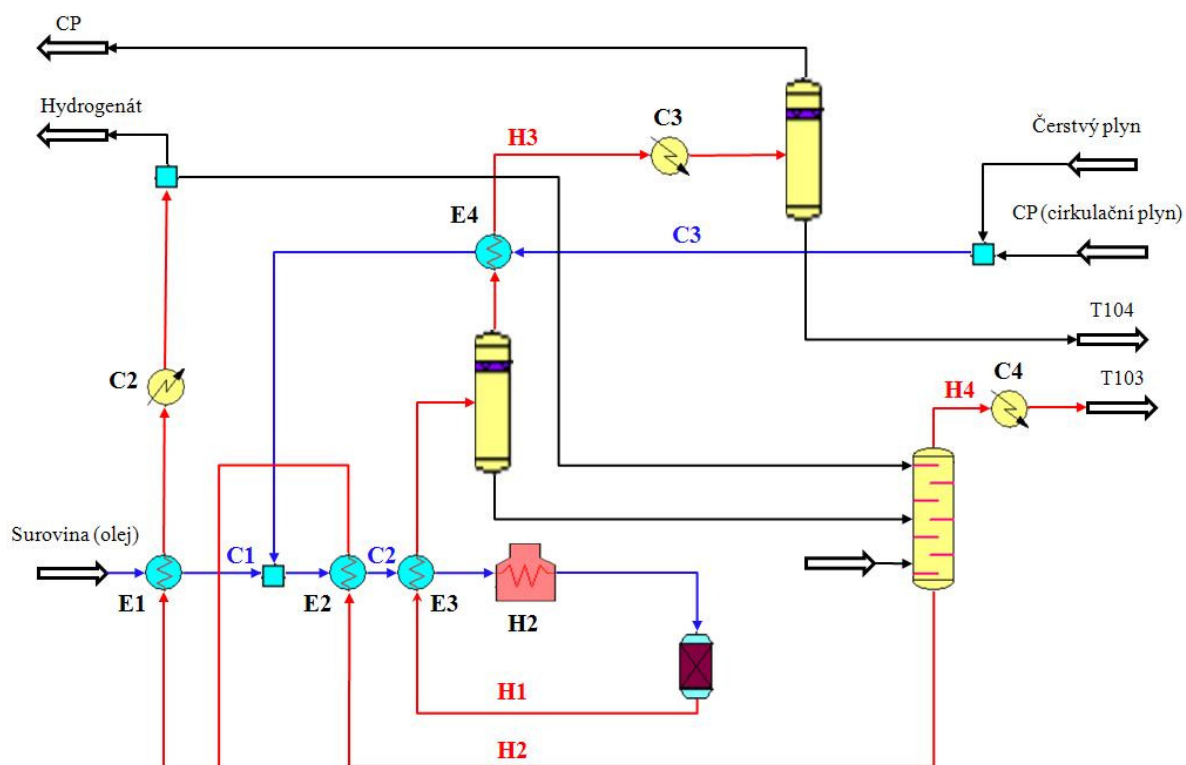


Obr. 25 Mřížková reprezentace hydrogenační rafinace oleje s prakticky upravenými výsledky deterministického modelu

Takto upravená (výsledná) mřížková reprezentace (Obr. 25) je následně překreslena do výsledného technologického schématu hydrogenační rafinace (Obr. 26).

Maximální procesní výměna tepla (MER) původního návrhu byla 1408,56 kW, výkon horkého servisu byl 1364,0 kW a celkový výkon chlazeného servisu byl 1816,3 kW. Výsledné řešení výměníkové sítě, provedené na základě použití deterministického modelu, dosahuje maximální procesní výměny tepla (MER) 1679,6 kW, výkonu horkého servisu 1091,0 kW a celkového výkonu chlazeného servisu 1563,4 kW.

Aplikací deterministického modelu tedy bylo obdrženo řešení, jenž vykazuje oproti původnímu projekčnímu návrhu zvýšený výkon výměníkové sítě o 19,3 %, přičemž nároky na horký servis dosáhnou 80,0 % původní projekční hodnoty a nároky na chladný servis dosáhnou 86,1 % původní projekční hodnoty. Obdržžený výsledek tak potvrzuje velmi dobré aplikační možnosti sestaveného deterministického modelu.



Obr. 26 Výsledné technologické schéma hydrogenační rafinace oleje s výměnou tepla navrženou pomocí deterministického modelu

9 Závěr

Cílem této diplomové práce bylo seznámení se s metodami pro syntézu sítě výměny tepla, s důrazem na dvě metody, a to na velmi známou *Pinch Design Method* a v poslední době dosti progresivní metodu matematického programování, označovanou jako tzv. deterministický model.

Po úvodu, v němž je uveden stručný přehled historického vývoje metod pro řešení syntézy sítě výměny tepla následuje podrobnější objasnění principů použití doposud velmi oblíbené *Pinch Design Method* a její postup řešení byl názorně prezentován na konkrétním příkladu. Byl zde také představen jeden z nadstavbových pomocných nástrojů pro řešení sítě výměny tepla touto metodou, tzv. Diagram hnací síly, který je velmi kvalitním doplňkem, zejména pro minimalizaci celkové plochy výměny tepla.

Těžištěm prací vykonaných v rámci této diplomové práce spočívalo ve vytvoření programové implementace deterministického modelu pro návrh sítě výměny tepla v prostředí softwaru *Maple*.

V průběhu vývoje a testování byly identifikovány následující prvky deterministického modelu ovlivňující kvalitu výsledné výměňkové sítě:

- Obecně publikovaný tvar deterministického modelu v literatuře [4] neobsahuje explicitní formulaci pro určení pořadí propojování procesních proudů pomocí výměníků v rámci jedné opakovací jednotky. Pořadí propojování procesních proudů má však, jak bylo zjištěno testovacími výpočty, významný dopad na kvalitu výsledné výměňkové sítě, zejména v případech nízkého počtu opakovacích jednotek malého rozsahu (málo procesních proudů účastnících se výměny tepla). Proto byla ve vytvořené programové implementaci použita sekvence propojování proudů (viz znázornění na *obr. 17*), jež byla na základě výsledků testovacích výpočtů identifikována pro tyto případy jako optimální.
- Volba rozsahu maximálních (parametr G) a minimálních (parametr ϵ) dovolených výkonů výměníků má významný vliv na kvalitu (osazení) výměňkové sítě.

Aplikace vytvořené implementace deterministického modelu je demonstrována na řešení tří teoretických a jednom konkrétním příkladu z praxe. Bylo zjištěno, že výsledky teoretických příkladů obdržené použitím deterministického modelu jsou rovnocenné s publikovanými výsledky těchto příkladů obdrženými pomocí *Pinch Design Method*.

V případě praktického příkladu se jednalo o proces hydrogenační rafinace oleje. Stávající uspořádání výměňkové sítě navržené projekční organizací svou vlastní (blíže nespecifikovanou) metodou bylo srovnáváno s návrhem provedeným pomocí deterministického modelu. Výsledky obdržené na základě deterministického modelu potvrdily jeho velmi dobré aplikační možnosti. Oproti původnímu projekčnímu návrhu bylo dosaženo zvýšení výkonu výměňkové sítě o 19,3 %, přičemž nároky na horký servis byly sníženy na 80,0 % a nároky na chladný servis na 86,1 % původní projekční hodnoty.

Seznam použitých zdrojů

- [1] STEHLÍK, P.: *Integrace procesů a její význam pro redukci spotřeby energie a škodlivých emisí -základní principy*, Nakladatelství "Procesní inženýrství", edice MAPRINT, Praha, 1995.
- [2] LINNHOFF, B., et al.: *User Guide on Process Integration for the Efficient Use of Energy*, IChemE, Rugby, U.K. , latest updated reprint, 1994
- [3] SMITH, R.: *Chemical Process Design*, McGraw-Hill, 1995
- [4] ERRICO, M., et al.: *A deterministic algorithm for the synthesis of maximum energy recovery heat exchanger network*, Computers and Chemical Engineering 31, pp.773–781, 2007
- [5] HEWITT, G.F., SHIRES, G.L., BOTT, T.R.: *Process Heat Transfer*, CRC Press, New York, USA, 1994
- [6] AHMAD, S., SMITH, R.: *Targets and design for minimum number of shell in heat exchanger network*, Chem. Eng. Res. Dev., vol. 67, pp. 481-494, 1989
- [7] JEGLA, Z.: *Studijní opory k předmětu „Systémové inženýrství III“*, Ústav procesního a ekologického inženýrství, VUT v Brně, 2009

Příloha 1

Procedury pro automatizované generování rovnic (výpis souboru: HENExprGen.mpl)

```
#####
#          PROCEDURE PRO AUTOMATICKÉ GENEROVÁNÍ ROVNIC PRO LP          #
#          OPTIMALIZAČNÍ MODEL SÍTĚ VÝMĚNÍKŮ TEPLA                  #
#          (PROCEDURES FOR AUTOMATIC GENERATION OF EXPRESSIONS FOR AN LP #
#          OPTIMIZATION MODEL OF A HEAT EXCHANGER NETWORK)           #
#                                                                      #
#          vysoké učení technické v Brně                             #
#          (Brno University of Technology)                             #
#                                                                      #
#          Fakulta strojního inženýrství                             #
#          (Faculty of mechanical engineering)                         #
#                                                                      #
#          Ústav procesního a ekologického inženýrství              #
#          (Institute of process and environmental engineering)        #
#          -----                                                  #
#          #
#          PROCEDURY (PROCEDURS)                                     #
#          #
#          generateConstraints(nHotStreams, nColdStreams, nRepUnits,  #
#          dTmin, M, Ths, Tht, Tcs, Tct, Cph, Cpc, maxDuty, minDuty,  #
#          maxExchangerCount):                                       #
#          Generuje omezení pro dané vstupní data                    #
#          (Generates constraints for the given input data).          #
#          VÝSTUP (OUTPUT): nastavení omezení (set of constraints.)   #
#          PARAMETRY (PARAMETERS):                                    #
#          nHotStreams      ... počet horkých proudů, [-]            #
#          (number of hot streams, [-])                               #
#          nColdStreams     ... počet chladných proudů, [-]         #
#          (number of cold streams, [-])                             #
#          nRepUnits        ... počet opakovacích jednotek, [-]     #
#          (number of repetitive units, [-])                         #
#          dTmin            ... minimální teplotní rozdíl, [°C]      #
#          minimum temperature difference, [°C]                      #
#          M                ... dostatečně malé záporné číslo k vynulování #
#          nerovnosti omezení                                       #
#          (negative value small enough to annul                     #
#          an inequality constraint, [°C])                            #
#          Ths              ... vstupní teplota horkého proudu, [°C] #
#          (supply temperature of hot stream (array), [°C])         #
#          Tht              ... výstupní teplota horkého proudu, [°C] #
#          (target temperature of hot stream (array), [°C])         #
#          Tcs              ... vstupní teplota chladného proudu, [°C] #
#          (supply temperature of cold stream (array), [°C])        #
#          Tct              ... výstupní teplota chladného proudu, [°C] #
#          (target temperature of cold stream (array), [°C])        #
#          Cph              ... tepelná kapacita horkého proudu, [kW/°C] #
#          (flowing heat capacities of hot streams                   #
#          (array), [kW/°C])                                         #
#          Cpc              ... tepelná kapacita chladného proudu, [kW/°C] #
#          (flowing heat capacities of cold streams                 #
#          (array), [kW/°C])                                         #
#          maxDuty          ... maximální výkon výměníku tepla, [kW] #
#          (maximum feasible heat duties of exchangers              #
#          (array), [kW])                                            #
#          minDuty          ... minimální výkon výměníku tepla, [kW] #
#          (minimum feasible heat duties of exchangers              #
#          (array), [kW])                                            #
#          maxExchangerCount ... maximální počet výměníků tepla, [-] #
#          (maximum number of heat exchangers)                      #
#          LOKÁLNÍ PROMĚNNÉ (LOCAL VARIABLES):                       #
#          coldStream       ... číslo chladného proudu protékajícího skrz #
#          daný výměník tepla, [-]                                   #
#          (number of the cold stream that flows through             #
#          a given heat exchanger, [-])                              #
#          constraints       ... nastavení omezení (set of constraints) #
#          hotStream        ... číslo horkého proudu protékajícího skrz #
#          daný výměník tepla, [-]                                   #
#          #
```

```

#           (number of the hot stream that flows through      #
#           a given heat exchanger, [-])                      #
#           nHeatExchangers ... celkový počet výměníků ve všech opakovacích jednotkách, [-] #
#           (total number of heat exchangers in all          #
#           repetitive units, [-])                            #
#           nofExchanger ... pomocné proměnné užívané během inicializace hotStream a coldStream, [-] #
#           (auxiliary variable used during initialization    #
#           of hotStream and coldStream arrays, [-])          #
#           i, j, k ... dolní indexi značí i-tý, j-tý proud nebo k-tý výměník tepla #
#           (subscripts denoting quantities related to the   #
#           i-th, j-th, or k-th stream or heat exchanger)     #
#           GLOBÁLNÍ PROMĚNNÉ (GLOBAL VARIABLES):             #
#           C ... tepelný výkon chladného servisu, [kw]      #
#           (cold utility, [kw])                              #
#           delta ... binární proměnná označující přítomnost daného výměníku #
#           0 - výměník neexistuje, 1 - výměník neexistuje), [-] #
#           (binary variable denoting presence of a given    #
#           heat exchanger (0 - exchanger is present,        #
#           1 - exchanger is not present), [-])              #
#           H ... tepelný výkon horkého servisu, [kw]        #
#           (hot utility, [kw])                               #
#           X ... tepelný výkon výměníku tepla, [kw]         #
#           heat duty of a heat exchanger, [kw]              #
#
# objFcnMEROnly(nHotStreams, nColdStreams, nRepUnits):      #
#   Generuje účelovou funkci (Generates objective function). #
#   VÝSTUP (OUTPUT): výraz reprezentující účelovou funkci  #
#   (expression representing objective function).            #
#   PARAMETRY (PARAMETERS):                                  #
#   nHotStreams ... počet horkých proudů, [-]                #
#   (number of hot streams, [-])                             #
#   nColdStreams ... počet chladných proudů, [-]            #
#   (number of cold streams, [-])                            #
#   nRepUnits ... počet opakovacích jednotek, [-]           #
#   (number of repetitive units, [-])                        #
#   LOKÁLNÍ PROMĚNNÉ (LOCAL VARIABLES):                      #
#   i ... dolní index označující tepelný výkon související s i-tým výměníkem tepla #
#   (subscript denoting heat duty related to the i-th heat exchanger) #
#   GLOBÁLNÍ PROMĚNNÉ (GLOBAL VARIABLES):                   #
#   X ... tepelný výkon výměníku tepla, [kw]                #
#   (heat duty of a heat exchanger, [kw])                   #
#####
generateConstraints:=proc(nHotStreams,nColdStreams,nRepUnits,dTmin,M,
                        Ths,Tht,Tcs,Tct,Cph,Cpc,maxDuty,minDuty,
                        maxExchangerCount)::set;
local constraints,i,j,k,hotStream,coldStream,nHeatExchangers,nofExchanger;
global X,C,H,delta;
description "Generates constraints for the given input data.";
constraints:={};
if (isInputSane(nHotStreams,nColdStreams,nRepUnits,dTmin,M,
                Ths,Tht,Tcs,Tct,Cph,Cpc,maxDuty,minDuty,
                maxExchangerCount)) then

# === Inicializace (Initialization) =====

nHeatExchangers::integer;
nHeatExchangers:=nRepUnits*nHotStreams*nColdStreams;
hotStream:=Array(1..nHeatExchangers,[],datatype=integer);
coldStream:=Array(1..nHeatExchangers,[],datatype=integer);
i:='i': j:='j': k:='k':
printlevel:=3:
for i from 1 to nColdStreams do
  for j from 1 to nHotStreams do
    for k from 1 to nRepUnits do
      nofExchanger:=nHotStreams*((k-1)*nColdStreams+i)-j+1;

```

```

        hotStream[nOfExchanger]:=j;
        coldStream[nOfExchanger]:=i;
    end do;
end do;
end do;

# === Rovnostní omezení (Equality constraints) =====
i:='i': j:='j':
for i from 1 to nHotStreams do
    constraints:=constraints union {Cph[i]*(Ths[i]-Tht[i])-
        add(piecewise(hotStream[j]=i,X[j],0),j=1..nHeatExchangers)-
        C[i]=0};
end do;
i:='i': j:='j':
for i from 1 to nColdStreams do
    constraints:=constraints union {Cpc[i]*(Tct[i]-Tcs[i])-
        add(piecewise(coldStream[j]=i,X[j],0),j=1..nHeatExchangers)-
        H[i]=0};
end do;
# === Logická omezení (Logical constraints) =====

i:='i':
for i from 1 to nHeatExchangers do
    constraints:=constraints union {maxDuty[i]*(1-delta[i])-X[i]>=0};
    constraints:=constraints union {X[i]+minDuty[i]*(delta[i]-1)>=0};
end do;

constraints:=constraints union
    {maxExchangerCount-add(1-delta[i],i=1..nHeatExchangers)>=0}:
# === Nerovnostní omezení (Inequality constraints) =====
# Horká strana výměníku (Hot sides of exchangers)
for i from 1 to nHeatExchangers do
    constraints:=constraints union {(C[hotStream[i]]+
        add(piecewise(hotStream[j]=hotStream[i],X[j],0),j=1..i))/
        Cph[hotStream[i]]-add(piecewise(coldStream[j]=coldStream[i],X[j],
        0),j=1..i)/Cpc[coldStream[i]]-dTmin-delta[i]*M+Tht[hotStream[i]]-
        Tcs[coldStream[i]]>=0};
end do;
# Chladná strana výměníku (Cold sides of exchangers)
for i from 1 to nHeatExchangers do
    constraints:=constraints union {(C[hotStream[i]]+
        add(piecewise(hotStream[j]=hotStream[i],X[j],0),j=1..i-1))/
        Cph[hotStream[i]]-add(piecewise(coldStream[j]=coldStream[i],X[j],
        0),j=1..i-1)/Cpc[coldStream[i]]-dTmin-delta[i]*M+
        Tht[hotStream[i]]-Tcs[coldStream[i]]>=0};
end do;
else
    ERROR("wrong input data, cannot proceed!");
end if;
return constraints;
end proc;

#####

isInputsSane:=proc(nHotStreams,nColdStreams,nRepUnits,dTmin,M,
    Ths,Tht,Tcs,Tct,Cph,Cpc,maxDuty,minDuty,
    maxExchangerCount)::boolean;

    local issane;
    description "Checks whether the input is sane.";
    issane::boolean:
    issane:=true;
    if (nHotStreams<=0) then
        ERROR("Number of hot streams cannot be less than one!");
    end if;
    if (nColdStreams<=0) then
        ERROR("Number of cold streams cannot be less than one!");
    end if;
    if (nRepUnits<=0) then
        ERROR("Number of repetitive units cannot be less than one!");
    end if;
    if (maxExchangerCount<=0) then
        ERROR("Maximum allowed exchanger count cannot be less than one!");
    end if;
    if (maxExchangerCount>(nHotStreams*nColdStreams*nRepUnits)) then

```

```
WARNING("maximum allowed exchanger count is set to a value "||
"greater than the maximum possible heat exchanger count.");
end if;
if (dTmin<=0) then
  ERROR("Minimum temperature difference must be greater than zero!");
end if;
if (M>=0) then
  ERROR("Constant M must be less than zero!");
end if;
if (ArrayNumElems(Ths)<>nHotStreams) then
  ERROR("Number od elements in the array representing supply "||
"temperatures of hot streams is different from the number "||
"of hot streams!");
end if;
if (ArrayNumElems(ThT)<>nHotStreams) then
  ERROR("Number od elements in the array representing target "||
"temperatures of hot streams is different from the number "||
"of hot streams!");
end if;
if (ArrayNumElems(Tcs)<>nColdStreams) then
  ERROR("Number od elements in the array representing supply "||
"temperatures of cold streams is different from the number "||
"of cold streams!");
end if;
if (ArrayNumElems(TcT)<>nColdStreams) then
  ERROR("Number od elements in the array representing target "||
"temperatures of cold streams is different from the number "||
"of cold streams!");
end if;
if (ArrayNumElems(Cph)<>nHotStreams) then
  ERROR("Number od elements in the array representing flowing heat "||
"capacities of hot streams is different from the number "||
"of hot streams!");
end if;
if (ArrayNumElems(Cpc)<>nColdStreams) then
  ERROR("Number od elements in the array representing flowing heat "||
"capacities of cold streams is different from the number "||
"of cold streams!");
end if;
if (ArrayNumElems(maxDuty)<>(nHotStreams*nColdStreams*nRepUnits)) then
  ERROR("Number od elements in the array representing maximum allowed "||
"heat duties is different from the maximum possible number "||
"of heat exchangers!");
end if;
if (ArrayNumElems(minDuty)<>(nHotStreams*nColdStreams*nRepUnits)) then
  ERROR("Number od elements in the array representing minimum allowed "||
"heat duties is different from the maximum possible number "||
"of heat exchangers!");
end if;
return issane;
end proc;

#####

objFcnMEROnly:=proc(nHotStreams,nColdStreams,nRepUnits)
  local i;
  global X;
  description "Generates objective function for simple ER maximization.";
  return add(X[i],i=1..nHotStreams*nColdStreams*nRepUnits);
end proc;

### KONEC (END OF FILE) #####
```

Příloha 2

Výsledné řešení příkladu pro ukázkou aplikace deterministické metody – 1 opakovací jednotka, 4 výměníky (výpis souboru: hen_optimization.UkazkaDetrmin.Metody.mws)

Optimalizace sítě výměny tepla

VUT v Brně, Fakulta strojního inženýrství
Ústav procesního a ekologického inženýrství

Seznam použitých identifikátorů

<i>C</i>	tepelný výkon chladného servisu, [kW]
<i>omezeni</i>	omezení
<i>Cpc</i>	tepelná kapacita chladného proudu, [kW/°C]
<i>Cph</i>	tepelná kapacita horkého proudu, [kW/°C]
<i>delta</i>	binární proměnná označující přítomnost daného výměníku tepla (0 - výměník existuje, 1 - výměník neexistuje), [-]
<i>dTmin</i>	minimální teplotní rozdíl, [°C]
<i>H</i>	tepelný výkon horkého servisu, [kW]
<i>M</i>	manipulační konstanta teplotního rozdílu (dostatečně malé záporné číslo k anulování omezení typu nerovnost), [°C]
<i>maxDuty</i>	maximální výkon výměníku tepla, [kW]
<i>maxExchangerCount</i>	maximální počet výměníků tepla, [-]
<i>minDuty</i>	minimální výkon výměníku tepla, [kW]
<i>nColdStreams</i>	počet chladných proudů [-]
<i>nHotStreams</i>	počet horkých proudů, [-]
<i>nRepUnits</i>	počet opakovacích jednotek, [-]
<i>ucelovaFunkce</i>	účelová funkce
<i>vysledek</i>	optimální řešení
<i>Tcs</i>	vstupní teplota chladného proudu, [°C]
<i>Tct</i>	výstupní teplota chladného proudu, [°C]
<i>Ths</i>	vstupní teplota horkého proudu, [°C]
<i>Tht</i>	výstupní teplota horkého proudu, [°C]
<i>X</i>	tepelný výkon výměníku tepla, [kW]

Vstupní hodnoty

```
> restart:
read "HENExprGen.mpl";
# === Deklarování typu proměnných =====
nHotStreams::integer:
nColdStreams::integer:
nRepUnits::integer:
dTmin::float:
M::float:
maxExchangerCount::integer:
> # === Vstupní data =====
nHotStreams:=2;
nColdStreams:=2;
nRepUnits:=1;
dTmin:=20;
M:=-1000;
Ths:=Array([300, 200], datatype=float):
Tht:=Array([80, 40], datatype=float):
Tcs:=Array([40, 140], datatype=float):
Tct:=Array([180, 280], datatype=float):
Cph:=Array([30, 45], datatype=float):
```

```
Cpc:=Array([40, 60],datatype=float):
maxDuty:=Array([4000,4000,4000,4000],datatype=float);
minDuty:=Array([500,500,500,500],datatype=float);
maxExchangerCount:=4;
```

```
nHotStreams :=2
nColdStreams :=2
nRepUnits :=1
dTmin :=20
M := -100C
maxDuty :=[ 4000. 4000. 4000. 4000. ]
minDuty :=[ 500. 500. 500. 500. ]
maxExchangerCount :=4
```

Omezení a účelová funkce

```
> # === Omezení =====
omezeni:=generateConstraints(nHotStreams,nColdStreams,nRepUnits,dTmin,M,Ths
,Tht,Tcs,Tct,Cph,Cpc,maxDuty,minDuty,maxExchangerCount);
```

$$\begin{aligned} \text{omezeni} := & \{ 6600. - X_2 - X_4 - C_1 = 0, 7200. - X_1 - X_3 - C_2 = 0, \\ & 5600. - X_1 - X_2 - H_1 = 0, 8400. - X_3 - X_4 - H_2 = 0, 0 \leq 4000 \\ & - 4000. \delta_1 - X_1, 0 \leq X_1 + 500. \delta_1 - 500., 0 \leq 4000. - 4000. \delta_2 \\ & - X_2, 0 \leq X_2 + 500. \delta_2 - 500., 0 \leq 4000. - 4000. \delta_3 - X_3, 0 \\ & \leq X_3 + 500. \delta_3 - 500., 0 \leq 4000. - 4000. \delta_4 - X_4, 0 \leq X_4 \\ & + 500. \delta_4 - 500., 0 \leq \delta_1 + \delta_2 + \delta_3 + \delta_4, 0 \leq 0.022222222222C_2 \\ & - 0.0027777777778X_1 - 20. + 1000 \delta_1, 0 \leq 0.033333333333C_1 \\ & + 0.008333333333X_2 - 0.025000000000X_1 + 20. + 1000 \delta_2, 0 \\ & \leq 0.022222222222C_2 + 0.022222222222X_1 + 0.005555555555X_3 \\ & - 120. + 1000 \delta_3, 0 \leq 0.033333333333C_1 + 0.033333333333X_2 \\ & + 0.016666666666X_4 - 0.016666666667X_3 - 80. + 1000 \delta_4, 0 \\ & \leq 0.022222222222C_2 - 20. + 1000 \delta_1, 0 \leq 0.033333333333C_1 \\ & - 0.025000000000X_1 + 20. + 1000 \delta_2, 0 \leq 0.022222222222C_2 \\ & + 0.022222222222X_1 - 120. + 1000 \delta_3, 0 \leq 0.033333333333C_1 \\ & + 0.033333333333X_2 - 0.016666666667X_3 - 80. + 1000 \delta_4 \} \end{aligned}$$

```
> # === Účelová funkce =====
ucelovaFunkce:=objFcnMEROnly(nHotStreams,nColdStreams,nRepUnits);
ucelovaFunkce :=X1 + X2 + X3 + X4
```

Výsledky

```
> with(Optimization):
vysledek:=LPSolve(ucelovaFunkce,omezeni,assume=nonnegative,binaryvariables=
{seq(delta[i],i=1..nHotStreams*nColdStreams*nRepUnits)},maximize);
```

$$\begin{aligned} \text{vysledek} := & [9999.9999992203 [X_4 = 2599.99999976025628, H_1 = 0., C_2 \\ & = 1400.00000053896724, X_2 = 1599.9999999896796, C_1 \\ & = 2400.00000024077462, X_1 = 4000.00000000103228, H_2 \\ & = 4000.00000077974346, X_3 = 1799.99999946000070, \delta_1 = 0, \delta_2 \\ & = 0, \delta_3 = 0, \delta_4 = 0]] \end{aligned}$$

Příloha 3

Výsledné řešení příkladu 1 – 2 opakovací jednotky, 5 výměníků (výpis souboru: hen_optimization.Příklad1.mws)

Optimalizace sítě výměny tepla

VUT v Brně, Fakulta strojního inženýrství
Ústav procesního a ekologického inženýrství

Seznam použitých identifikátorů

<i>C</i>	tepelný výkon chladného servisu, [MW]
<i>omezeni</i>	omezení
<i>Cpc</i>	tepelná kapacita chladného proudu, [MW/°C]
<i>Cph</i>	tepelná kapacita horkého proudu, [MW/°C]
<i>delta</i>	binární proměnná označující přítomnost daného výměníku tepla (0 - výměník existuje, 1 - výměník neexistuje), [-]
<i>dTmin</i>	minimální teplotní rozdíl, [°C]
<i>H</i>	tepelný výkon horkého servisu, [MW]
<i>M</i>	manipulační konstanta teplotního rozdílu (dostatečně malé záporné číslo k anulování omezení typu nerovnost), [°C]
<i>maxDuty</i>	maximální výkon výměníku tepla, [MW]
<i>maxExchangerCount</i>	maximální počet výměníků tepla, [-]
<i>minDuty</i>	minimální výkon výměníku tepla, [MW]
<i>nColdStreams</i>	počet chladných proudů [-]
<i>nHotStreams</i>	počet horkých proudů, [-]
<i>nRepUnits</i>	počet opakovacích jednotek, [-]
<i>ucelovaFunkce</i>	účelová funkce
<i>vysledek</i>	optimální řešení
<i>Tcs</i>	vstupní teplota chladného proudu, [°C]
<i>Tct</i>	výstupní teplota chladného proudu, [°C]
<i>Ths</i>	vstupní teplota horkého proudu, [°C]
<i>Tht</i>	výstupní teplota horkého proudu, [°C]
<i>X</i>	tepelný výkon výměníku tepla, [MW]

Vstupní hodnoty

```
> restart:
read "HENExprGen.mpl";
# === Deklarování typu proměnných =====
nHotStreams::integer:
nColdStreams::integer:
nRepUnits::integer:
dTmin::float:
M::float:
maxExchangerCount::integer:
> # === Vstupní data =====
nHotStreams:=2;
nColdStreams:=2;
nRepUnits:=2;
dTmin:=10;
M:=-1000;
Ths:=Array([250, 200],datatype=float):
Tht:=Array([40, 80],datatype=float):
Tcs:=Array([20, 140],datatype=float):
Tct:=Array([180, 230],datatype=float):
Cph:=Array([0.15, 0.25],datatype=float):
Cpc:=Array([0.2, 0.3],datatype=float):
```

```
maxDuty:=Array([20,20,20,20,20,20,20,20],datatype=float);
minDuty:=Array([5,5,5,5,5,5,5,5],datatype=float);
maxExchangerCount:=5;
```

```

nHotStreams :=2
nColdStreams :=2
nRepUnits :=2
dTmin :=10
M := -100C
maxDuty :=[ 20. 20. 20. 20. 20. 20. 20. 20. ]
minDuty :=[ 5. 5. 5. 5. 5. 5. 5. 5. ]
maxExchangerCount :=5
```

Omezení a účelová funkce

```
> # === Omezení =====
omezeni:=generateConstraints(nHotStreams,nColdStreams,nRepUnits,dTmin,M,Ths
, Tht, Tcs, Tct, Cph, Cpc, maxDuty, minDuty, maxExchangerCount):
#výpis množiny omezení potlačen z důvodu rozsáhlosti

> # === Účelová funkce =====
ucelovaFunkce:=objFcnMEROnly(nHotStreams,nColdStreams,nRepUnits);
ucelovaFunkce :=X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8
```

Výsledky

```
> with(Optimization):
vysledek:=LPSolve(ucelovaFunkce,omezeni,assume=nonnegative,binaryvariables=
{seq(delta[i],i=1..nHotStreams*nColdStreams*nRepUnits)},maximize);
```

```
vysledek :=[51.500000000939[δ5 = 0, δ6 = 0, δ7 = 0, δ8 = 0, X2
= 6.49999999959844298X4 = 3.441992619333332870-10, X6
= 7.99999999977052756X8 = 7.00000000088236796C1
= 9.9999999994044604X1 = 0., X3 = 1.157119430749550180-14,
X5 = 17.5000000006310296X7 = 12.4999999997131610C2 = 0., H1
= 0., δ1 = 1, δ2 = 0, δ3 = 1, δ4 = 1, H2 = 7.4999999990602583}}
```

Příloha 4

Výsledné řešení příkladu 1 – 2 opakovací jednotky, 6 výměníků (výpis souboru: hen_optimization.Příklad2.mws)

Optimalizace sítě výměny tepla

VUT v Brně, Fakulta strojního inženýrství
Ústav procesního a ekologického inženýrství

Seznam použitých identifikátorů

<i>C</i>	tepelný výkon chladného servisu, [MW]
<i>omezeni</i>	omezení
<i>Cpc</i>	tepelná kapacita chladného proudu, [MW/°C]
<i>Cph</i>	tepelná kapacita horkého proudu, [MW/°C]
<i>delta</i>	binární proměnná označující přítomnost daného výměníku tepla (0 – výměník existuje, 1 - výměník neexistuje), [-]
<i>dTmin</i>	minimální teplotní rozdíl, [°C]
<i>H</i>	tepelný výkon horkého servisu, [MW]
<i>M</i>	manipulační konstanta teplotního rozdílu (dostatečně malé záporné číslo k anulování omezení typu nerovnost), [°C]
<i>maxDuty</i>	maximální výkon výměníku tepla, [MW]
<i>maxExchangerCount</i>	maximální počet výměníků tepla, [-]
<i>minDuty</i>	minimální výkon výměníku tepla, [MW]
<i>nColdStreams</i>	počet chladných proudů [-]
<i>nHotStreams</i>	počet horkých proudů, [-]
<i>nRepUnits</i>	počet opakovacích jednotek, [-]
<i>ucelovaFunkce</i>	úcelová funkce
<i>vysledek</i>	optimální řešení
<i>Tcs</i>	vstupní teplota chladného proudu, [°C]
<i>Tct</i>	výstupní teplota chladného proudu, [°C]
<i>Ths</i>	vstupní teplota horkého proudu, [°C]
<i>Tht</i>	výstupní teplota horkého proudu, [°C]
<i>X</i>	tepelný výkon výměníku tepla, [MW]

Vstupní hodnoty

```
> restart:
read "HENExprGen.mpl";
# === Deklarování typu proměnných =====
nHotStreams::integer:
nColdStreams::integer:
nRepUnits::integer:
dTmin::float:
M::float:
maxExchangerCount::integer:
> # === Vstupní data =====
nHotStreams:=2;
nColdStreams:=3;
nRepUnits:=3;
dTmin:=20;
M:=-1000;
Ths:=Array([145, 140],datatype=float):
Tht:=Array([60, 65],datatype=float):
Tcs:=Array([40, 80, 40],datatype=float):
Tct:=Array([120, 100, 100],datatype=float):
Cph:=Array([2, 4],datatype=float):
Cpc:=Array([3, 5, 1.5],datatype=float):
```

```
maxDuty:=Array([130,130,130,130,130,130,130,130,130,130,130,130,130,130,130,130,130,130],datatype=float);
minDuty:=Array([10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10],datatype=float);
maxExchangerCount:=6;
```

```

nHotStreams :=2
nColdStreams :=3
nRepUnits :=3
dTmin :=20
M := -100C
maxDuty := [ 1 .. 18 Array
              Data Type: float8
              Storage: rectangular
              Order: Fortran_order ]
minDuty := [ 1 .. 18 Array
              Data Type: float8
              Storage: rectangular
              Order: Fortran_order ]
maxExchangerCount :=6
```

Omezení a účelová funkce

```
> # === Omezení =====
omezeni:=generateConstraints(nHotStreams,nColdStreams,nRepUnits,dTmin,M,Ths,
, Tht, Tcs, Tct, Cph, Cpc, maxDuty, minDuty, maxExchangerCount):
#výpis množiny omezení potlačen z důvodu rozsáhlosti

> # === Účelová funkce =====
ucelovaFunkce:=objFcnMEROnly(nHotStreams,nColdStreams,nRepUnits);
ucelovaFunkce :=X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9
+ X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18
```

Výsledky

```
> with(Optimization):
vysledek:=LPSolve(ucelovaFunkce,omezeni,assume=nonnegative,binaryvariables=
{seq(delta[i],i=1..nHotStreams*nColdStreams*nRepUnits)},maximize);
vysledek := [430.00000000034 [X8 = 89.9999999301011542X2 = 0., X4
= 1.559090733554960760-10, X6 = 60.0000000531088276X10
= 0., X5 = 0., X12 = 0., X14 = 0., X16 = 2.701007417726943290-9,
X18 = 0., C1 = 20.0000000139330752X1 = 0., X3 = 0., X7
= 120.000000014402018X9 = 99.99999997487292X11
= 29.9999999468911654X13 = 30.0000000554968089X15 = 0., X17
= 0., C2 = 19.9999999857227202H1 = 0., H2 = 0., H3 = 0., δ1 = 1,
δ2 = 1, δ3 = 1, δ4 = 1, δ5 = 1, δ6 = 0, δ7 = 0, δ8 = 0, δ9 = 0, δ10 = 1,
δ11 = 0, δ12 = 1, δ13 = 0, δ14 = 1, δ15 = 1, δ16 = 1, δ17 = 1, δ18 = 1]]]
```

Příloha 5

Výsledné řešení příkladu 3 – 1 opakovací jednotky, 4 výměníků (výpis souboru: hen_optimization.Příklad3.mws)

Optimalizace sítě výměny tepla

VUT v Brně, Fakulta strojního inženýrství
Ústav procesního a ekologického inženýrství

Seznam použitých identifikátorů

<i>C</i>	tepelný výkon chladného servisu, [kW]
<i>omezeni</i>	omezení
<i>Cpc</i>	tepelná kapacita chladného proudu, [kW/°C]
<i>Cph</i>	tepelná kapacita horkého proudu, [kW/°C]
<i>delta</i>	binární proměnná označující přítomnost daného výměníku tepla (0 - výměník existuje, 1 - výměník neexistuje), [-]
<i>dTmin</i>	minimální teplotní rozdíl, [°C]
<i>H</i>	tepelný výkon horkého servisu, [kW]
<i>M</i>	manipulační konstanta teplotního rozdílu (dostatečně malé záporné číslo k anulování omezení typu nerovnost), [°C]
<i>maxDuty</i>	maximální výkon výměníku tepla, [kW]
<i>maxExchangerCount</i>	maximální počet výměníků tepla, [-]
<i>minDuty</i>	minimální výkon výměníku tepla, [kW]
<i>nColdStreams</i>	počet chladných proudů [-]
<i>nHotStreams</i>	počet horkých proudů, [-]
<i>nRepUnits</i>	počet opakovacích jednotek, [-]
<i>ucelovaFunkce</i>	účelová funkce
<i>vysledek</i>	optimální řešení
<i>Tcs</i>	vstupní teplota chladného proudu, [°C]
<i>Tct</i>	výstupní teplota chladného proudu, [°C]
<i>Ths</i>	vstupní teplota horkého proudu, [°C]
<i>Tht</i>	výstupní teplota horkého proudu, [°C]
<i>X</i>	tepelný výkon výměníku tepla, [kW]

Vstupní hodnoty

```
> restart:
read "HENExprGen.mpl";
# === Deklarování typu proměnných =====
nHotStreams::integer:
nColdStreams::integer:
nRepUnits::integer:
dTmin::float:
M::float:
maxExchangerCount::integer:
> # === Vstupní data =====
nHotStreams:=4;
nColdStreams:=3;
nRepUnits:=1;
dTmin:=73.6;
M:=-1000;
Ths:=Array([289.09, 213.38, 214.59, 144.42 ],datatype=float):
Tht:=Array([214.59, 70, 30, 30 ],datatype=float):
Tcs:=Array([77, 113.26, 95 ],datatype=float):
Tct:=Array([108, 289, 130 ],datatype=float):
Cph:=Array([13.74, 10.32, 2.59, 2.31 ],datatype=float):
Cpc:=Array([8.49, 13.59, 3.48 ],datatype=float):
```

```
maxDuty:=Array([1500,1500,1500,1500,1500,1500,1500,1500,1500,1500,1500,1500],datatype=float);
minDuty:=Array([100,100,100,100,100,100,100,100,100,100,100,100],datatype=float);
maxExchangerCount:=4;
```

```
nHotStreams:=4
nColdStreams:=3
nRepUnits:=1
dTmin:=73.6
M:=-100C
maxDuty:=
    [
        1..12 Array
        Data Type: float8
        Storage: rectangular
        Order: Fortran_order
    ]
minDuty:=
    [
        1..12 Array
        Data Type: float8
        Storage: rectangular
        Order: Fortran_order
    ]
maxExchangerCount:=4
```

Omezení a účelová funkce

```
> # === Omezení =====
omezeni:=generateConstraints(nHotStreams,nColdStreams,nRepUnits,dTmin,M,Ths
, Tht, Tcs, Tct, Cph, Cpc, maxDuty, minDuty, maxExchangerCount):
#výpis množiny omezení potlačen z důvodu rozsáhlosti

> # === Účelová funkce =====
ucelovaFunkce:=objFcnMEROnly(nHotStreams,nColdStreams,nRepUnits);
ucelovaFunkce :=X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9
+ X10 + X11 + X12
```

Výsledky

```
> with(Optimization):
vysledek:=LPSolve(ucelovaFunkce,omezeni,assume=nonnegative,binaryvariables=
{seq(delta[i],i=1..nHotStreams*nColdStreams*nRepUnits)},maximize);
```

```
vysledek := [1679.6205000473 [X4 = 0., X8 = 1023.62999845009540
X12 = 0.0000015499046656667541, X1 = 0., X3
= 263.18999999999998, X7 = 273.686400046752169, X11 = 0., C2
= 942.805199953247552, X2 = 0., X6 = 3.441992619333332870-10,
X10 = 119.114099998872546, C3 = 358.974000000783292, X1 = 0.,
X5 = 3.441992619333332870-10, X9 = 1.032597785799999970-9,
C4 = 264.310199998623204, H1 = 0., H2 = 1090.99020150246406
H3 = 2.685898450190186808, δ1 = 1, δ2 = 1, δ3 = 0, δ4 = 1, δ5 = 1, δ6
= 1, δ7 = 0, δ8 = 0, δ9 = 1, δ10 = 0, δ11 = 1, δ12 = 1]]]
```