REFORMULAÇÃO DO ALGORITMO DE SÍNTESE DE REDES DE TROCADORES DE CALOR NO SOFTWARE DIDÁTICO MPINCH

Lucas Soares Gaiola (PIC/UEM), Rodrigo Cesar Rodrigues (PIC/ UEM), Leandro Vitor Pavão (Autor), Mauro Antônio da Silva Sa Ravagnani(Autor), Camila de Brito Miranda Faia (Orientador), Caliane Bastos Borba Costa (Coorientador) e-mail: ra98768@uem.br, cbmiranda2@uem.br.

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá, PR.

Engenharias/ Engenharia Química

Palavras-chave: metodologia *Pinch*, superestrutura, linguagem Python

Resumo:

O presente trabalho apresenta os resultados da reformulação do código do programa MPinch, inicialmente escrito em MATLAB, para a linguagem Python com enfoque no algoritmo de cálculo da rede de trocadores de calor (RTC) utilizando modelos derivados de superestrutura. O programa original realiza o cálculo das temperaturas do ponto de estrangulamento energético (ponto *pinch*), através de uma cascata energética baseada nas energias das correntes que foram calculadas pelas leis da termodinâmica, visando a utilização do mesmo para calcular uma RTC com menor consumo de utilidades frias e quentes, de modo que não existe fluxo de energia atravessando o ponto *pinch*. Entretanto, o mesmo possuía falhas em alguns aspectos de modo a ser necessário sua reformulação. Os resultados obtidos são comparados com o programa original, avaliando sua velocidade de execução e a solução das antigas falhas presentes.

Introdução

Nos processos industriais há correntes quentes e frias, que são fluidos que, durante o processo, precisam ser resfriados ou aquecidos respectivamente. Esse processo de troca térmica pode ser realizado entre as próprias correntes por meio da passagem por um trocador de calor, trocando energia entre si, ou pode ser entre corrente e utilidades. Definem-se utilidades como correntes externas ao processo, utilizadas para fornecer ou remover calor. O conjunto dos referidos equipamentos para troca térmica forma uma rede de trocadores de calor (RTC) (RAVAGNANI; SUÁREZ, 2012).

Deve-se notar que a produção de utilidades possui um custo associado. Desta forma a metodologia *Pinch* tem como objetivo aproveitar ao máximo a energia já presente no processo contida nas correntes quentes e podendo ser recebida pelas frias. Minimiza-se, desta forma, o consumo de utilidades. Este método se desenvolve por meio de regras heurísticas e utilizando os princípios das leis da

termodinâmica para o cálculo das energias das correntes, possibilitando o cálculo do chamado “ponto de estrangulamento”, ou seja, da “temperatura *pinch*”. Isso se dá através de uma sequência lógica de cálculos, que partem da energia presente em cada corrente, para obter uma cascata de energia que fornecerá a temperatura desejada no ponto em que não há fluxo de energia. Dessa temperatura, divide-se a rede em duas regiões, acima do ponto e abaixo do ponto. Permite-se que utilidades frias sejam utilizadas apenas na região abaixo do ponto e utilidades quentes apenas acima do ponto (RAVAGNANI; SUÁREZ, 2012).

Este método, ainda que auxilie no desenvolvimento de uma RTC, ainda não é autônomo para tal, exigindo que o usuário faça manualmente parte do processo. De modo a automatizar a síntese de uma RTC, as técnicas de programação matemática podem ser utilizadas. O uso de tais abordagens, de modo geral, passa pela derivação de modelos matemáticos a partir de superestruturas que contemplam várias topologias para uma RTC. Ao se utilizarem técnicas de otimização, pode-se buscar uma solução para esse modelo que tenha associado um custo mínimo, considerando-se custos operacionais e de capital. É importante notar ainda que modelos derivados de superestruturas são robustos em sua programação, dada a sistematicidade de sua organização por meio de variáveis vetoriais e matriciais binárias e contínuas. Estas representam diversas características das redes, desde a existência ou ausência de um equipamento até a carga térmica do mesmo (PAVÃO; COSTA; RAVAGNANI, 2018).

Cabe ressaltar que o MPinch é um software para auxílio no uso da metodologia *Pinch*, que não depende de abordagens de otimização. Entretanto, em vista das características apresentadas a respeito de superestruturas, pode-se aproveitar as vantagens dos modelos derivados destas na programação da presente ferramenta. Em especial, a organização matricial das variáveis pode ser utilizada com fins de tornar a programação mais robusta, com menor probabilidade de ocorrência de *bugs*. Uma vez que os dados serão processados nesse formato, é facilitado ainda o acoplamento de abordagens de otimização ao programa, o que pode ser uma alternativa viável para projetos futuros.

A linguagem Python utilizada na reformulação do código é de simples utilização e entendimento, além de ser totalmente gratuita, diferente da linguagem anterior que era de um software proprietário (MATLAB). A nova linguagem permite que o programa possua um código menor, tornando-o mais leve e de rápida execução.

Sendo assim, a reformulação do *software* visa não só corrigir os *bugs* do programa anterior como acrescentar o método de superestruturas para o desenvolvimento da RTC, permitindo que os dados dos usuários sejam salvos para facilitar a ampliação de funções do *software*, como a otimização da rede.

Materiais e métodos

No início do trabalho uma revisão bibliográfica foi realizada, visando compreender a utilização da metodologia *Pinch* e de superestruturas no

desenvolvimento de uma RTC. Posteriormente foi realizada mais uma revisão bibliográfica para ambientação com a programação Python, para analisar possibilidades de utilização na reformulação do MPinch. Após isso, foi desenvolvido o código, iniciando com as rotinas de cálculo para encontrar o ponto *pinch* e seguindo com os cálculos relativos à superestrutura a partir do cálculo da temperatura *pinch*. Uma versão simplificada da superestrutura de Pavão, Costa e Ravagnani (2018) foi utilizada para a representação e armazenamento de dados de rede.

Resultados e Discussão

Utilizando como base os métodos algorítmicos da síntese de redes de trocadores de calor, implementou-se o código para o cálculo do ponto *pinch* de uma rede com qualquer número de correntes presentes. Em seguida, o projeto foi voltado para a parte visual do programa. Com as bibliotecas *matplotlib*, *numpy*, *pyqt5* e *xlrd*, foi possível criar uma interface simples e eficiente, como apresentado na Figura 1.

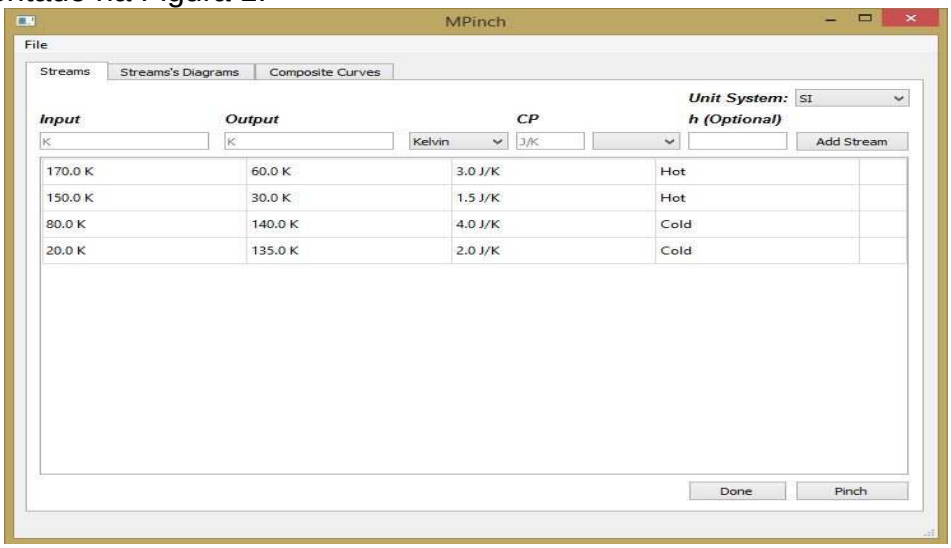

Figura 1 - Interface inicial do MPinch, com dados de temperatura e capacidade caloríficas já inseridos

Após feita a interface inicial, utilizou-se a versão simplificada da superestrutura de Pavão, Costa e Ravagnani (2018) (Figura 2) para a sistematização da síntese da rede de trocadores de calor em uma linguagem de programação de alto nível.

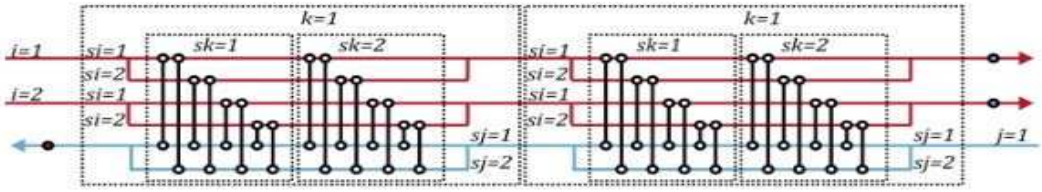

Figura 2 - Aplicação ilustrativa da superestrutura de Pavão, Costa e Ravagnani (2018).

A superestrutura desenvolvida contempla como opções estruturais divisões de correntes em sub-correntes e todas as possíveis trocas térmicas entre essas.

Há, na proposta, variáveis de decisão, a partir das quais pode-se calcular os balanços de energia da RTC. Estas são:

I. a variável binária representando existência/ausência de uma troca térmica, $z_{i,si,j,sj,sk,k}$
II. as cargas térmicas dos trocadores de calor, $Q_{i,si,j,sj,sk,k}$
III. as frações das divisões de correntes, $Fh_{i,si,k}$ e $Fc_{j,si,k}$

Os índices são *i* (corrente quente), *si* (sub-corrente quente), *j* (corrente fria), *sj* (sub-corrente fria), *sk* (sub-estágio) e *k* (estágio). Por exemplo, uma variável $z_{i,si,j,sj,sk,k}$ denota a existência de uma troca térmica entre a sub-corrente quente *si* pertencente à corrente quente *i* com a sub-corrente fria *sj*, pertencente à corrente fria *j* no sub-estágio *sk* pertencente ao estágio *k*. Uma variável de fração $Fh_{i,si,k}$ representa a fração da vazão que é desviada da corrente quente *i* para a sub-corrente quente *si* no estágio *k*. A Figura 2 ilustra os índices utilizados na representação da rede.

Para a transcrição do modelo matemático para a linguagem *Python*, utilizou-se a biblioteca *numpy*, que possibilitou a criação de uma matriz com 6 índices (correntes quentes e frias, subcorrentes quentes e frias, estágios e subestágios) que armazenam a coordenada de cada trocador presente na rede. Com isso, foi possível desenvolver o algoritmo que efetua o cálculo da troca térmica de qualquer trocador entre duas correntes.

Conclusões

Utilizando a linguagem *Python* juntamente com algumas bibliotecas, foi possível concluir que pode-se criar um programa capaz de realizar uma síntese de trocadores de calor de maneira direta eficiente. Também foi possível comprovar como o modelo matemático da superestrutura utilizada é eficiente para a aplicação prática e algorítmica da síntese de uma rede de trocadores de calor.

Agradecimentos

Os autores agradecem a UEM pelo apoio na realização do projeto no qual esse trabalho está inserido.

Referências

PAVÃO, L. V.; COSTA, C. B. B.; RAVAGNANI, M. A. S. S. A new stage-wise superstructure for heat exchanger network synthesis considering substages, sub-splits and cross flows. **Applied Thermal Engineering**, v. 143, P. 719-735, 2018.

RAVAGNANI, M. A. S. S.; SUÁREZ, J. A. C. A. **Redes de Trocadores de Calor**. 1. ed. Maringá: Eduem, 2012.