POLITECNICO DI MILANO

School of Industrial and Information Engineering

Department of Chemistry, Materials and Chemical Engineering

"Giulio Natta"

Master of Science in Chemical Engineering

Surrogate Model in PSE Context: Optimization of a Heat Exchanger Network

SuPER Team

Sustainable Process Engineering Research

Supervisor :

Prof. Flavio MANENTI

Co-supervisor :

Prof. Alessandro DI PRETORO

Candidate :

Valerio Lorenzo MUCI

Matr. n. 880438

Academic Year 2018–2019

INDEX

FIGURE INDEX

TABLE INDEX

ABSTRACT

In every industrial plant an efficient heat exchanger network (HEN) is mandatory to guarantee the temperature control and to optimize the energy requirement, thus reducing risks and operating costs. Therefore, chemical engineers have made a lot of researches in this direction as a good heat exchange can bring many advantages to the entire process, so it has gained a growing interest over the past decades. However, this work has two main objectives: the first is to find a feasible way of optimization of a HEN, like the Grossmann's superstructure; the latter is to apply an innovative technique to analyze and hopefully improve whatever system, that is surrogate modeling. In fact, this new informatic instrument elaborates some training points taken from real observations, then it generates an analytical function which approximates and describes those phenomena as best as possible. For this goal, in the first step the academic software MATLAB® supports simulation and optimization of the heat exchanger network; afterward, in the second step the results of the previous part have been sampled and some training sets have been implemented in the recent software ALAMO, which has been utilized to make a sort of constrained regression in order to create a surrogate model. The results of this work show that computational efforts of the optimization increase as the size of the heat exchanger network is progressively augmented; in addition, as the number of heat exchanger units is increased, it is difficult to analyze those results, because their mathematical dimensions are more than four. Finally, ALAMO has demonstrated a good reliability, even when applied to samples of different magnitude, thus resulting quite interesting.

RIASSUNTO

In ogni impianto industriale è obbligatoria un'efficiente rete di scambiatori di calore (HEN) per garantire il controllo della temperatura e ottimizzare il fabbisogno energetico, riducendo così i rischi e i costi operativi. Pertanto, gli ingegneri chimici hanno fatto molte ricerche in questa direzione poiché un buon scambio di calore può portare molti vantaggi all'intero processo, quindi ha acquisito un crescente interesse negli ultimi decenni. Tuttavia, questo lavoro ha due obiettivi principali: il primo è trovare un modo fattibile di ottimizzazione di una HEN, come la superstruttura di Grossmann; il secondo è quello di applicare una tecnica innovativa per analizzare e, auspicabilmente, migliorare qualsiasi sistema, cioè la modellazione surrogata (surrogate modeling). In effetti, questo nuovo strumento informatico elabora alcuni training points tratti da osservazioni reali, quindi genera una funzione analitica che approssima e descrive quei fenomeni nel miglior modo possibile. Per questo obiettivo, nella prima fase il software accademico MATLAB® supporta la simulazione e l'ottimizzazione della rete di scambiatori di calore; in seguito, nella seconda fase, i risultati della parte precedente sono stati campionati e alcuni training sets sono stati implementati nel recente software ALAMO, che è stato utilizzato per creare una sorta di regressione vincolata al fine di creare un modello surrogato. I risultati di questo lavoro mostrano che gli sforzi computazionali dell'ottimizzazione aumentano all'aumentare della dimensione della rete di scambiatori di calore; inoltre, poiché il numero di unità di scambiatore di calore è progressivamente aumentato, è difficile analizzare tali risultati, poiché le loro dimensioni matematiche sono più di quattro. Infine, ALAMO ha dimostrato una buona affidabilità, anche quando applicato a campioni di diversa ampiezza, risultando quindi piuttosto interessante.

1 INTRODUCTION

1.1 General overview

Over the last decades, the optimization of heat exchangers has become extremely important because of many factors. First of all, the number of industries and their complexity have increased a lot, thus causing larger and larger energy demands as well as more significant supply chain issues. Furthermore, other problems come from the political worldwide panorama, which continuously changes and has already demonstrated a lot of times that long stability is almost impossible in many places in the world. Without any doubt, a well-known example is the oil crisis during the '70, where a lot of European nations had to face the precarity of fossil fuels, which were the principal energy source for most processes until that period. From there on, people understood how important a good energy management and saving may be, so they started a lot of researches in this wide field. By improving this side of the production, operating costs as well as the dependence from external factors could be largely reduced, while the concept of ecology started to diffuse all around the world.

1.2 Motivation

Among this big amount of studies about heat exchange optimization, the work of the professor and researcher Ignacio Grossmann has been extremely interesting and it has given great hints to the whole scientific community. After having obtained his degree in Master Science and Ph.D of Chemical Engineering, he joined the Carnegie Mellon of Pittsburgh in 1979, then he rapidly became an influent researcher in the field of chemical process optimization. Moreover, he is member of many engineering associations and institutes, like the American Institute of Chemical engineering. One of his studies is just about the optimization of a superstructure of a Heat Exchangers Network (HEN), proposed in 1992 for the first time, whose scheme is reported below:

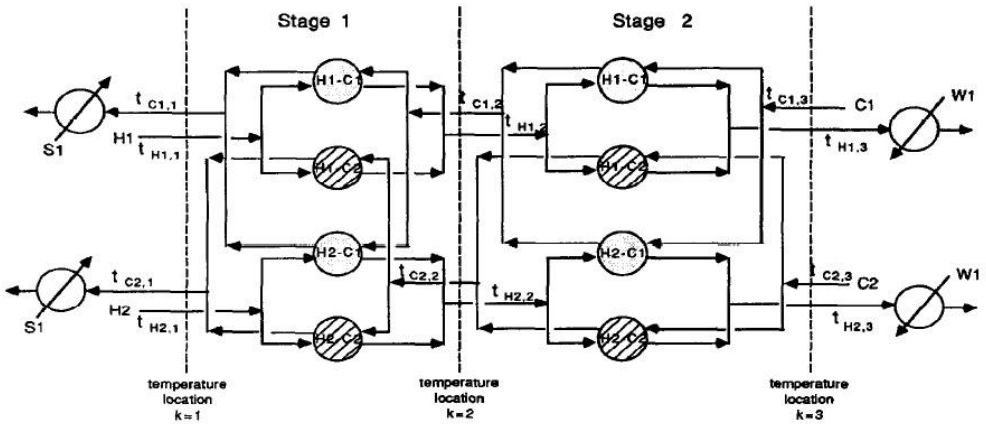


***Figure 1**. Grossmann's heat exchanger network superstructure. (Grossmann I. E., 1990)*

In general, a HEN synthesis problem may be addressed giving the pieces of information as follows:

- A set of hot process streams HP to be cooled and a set of cold process streams CP to be heated;
- Heat capacity flow rates of each hot and cold stream;
- Initial and target temperatures;
- A set of hot utilities HU and a set of cold utilities CU and their corresponding temperatures.

Of course, the objective is to find the HEN which presents the lowest annual cost and the solution may be computed by providing:

- Utilities required by the entire system;
- Stream matches and number of units as well as the resulting network configuration and the flows for all branches;
- Heat loads and operating temperatures for each heat exchanger;
- Exchange are of all the units.

In addition, some constraints on stream matches, stream splits and number of units may be imposed (Grossmann I. E., 1990).

However, in the specific case of Grossmann's HEN superstructure, there are two hot and cold streams along with hot and cold utilities; the number of stages in the superstructure has been set to the maximum number of hot or cold streams. For the sake of simplicity, are placed at the end of each flow and the consecutive mixing of fluxes is assumed to be isothermal and perfect, thus eliminating difficult nonlinear heat balances as well as nonlinear heat mixing equations and allowing the modeler to consider just an overall balance around each stage. So, the feasible space is defined by strictly linear constraints and the only nonlinearity is referred to the final cost estimation.

1.3 Objective

The aforesaid superstructure results to be globally a Mixed Integer Non-Linear Problem (MINLP), where both binary variables (passing or not passing) and continuous variables (while passing) must be all considered. This feature makes the problem quite difficult to solve even with the most modern software, as the required computational efforts are huge. In effect, the classical methods involving balance equations over each heat exchanger have a lot of problems, which they cannot completely solve still today. Hence, the main objective of this work is to search for a new solving path which is able to overcome those difficulties in the structure of the problem itself. That's why over the very last years surrogate modeling has been considered a good candidate to find a solution or at least to help approach it. Indeed, the most interesting characteristic of these innovative research field is the mathematical simplification of real systems, like heat exchanger networks or chemical processes. The main goal of surrogates is the informatic elaboration of data sets previously sampled and the

successive construction of an analytical function which describe as best as possible the subject phenomena. Furthermore, this new strategy has been progressively enforced by all improvements in power and efficiency of computers until today. For these reasons, some researchers have thought of the possibility to implement surrogate models in the heat exchanger optimization, as global computational efforts may be largely reduced and all the problems typical of each MINLP may be easily overcome.

In particular, the target of this thesis project is to prove and verify the reliability of such technique by means of the new software ALAMO (Alison Cozad, Automated Learning of Algebraic Models for Optimization, 2014), developed by professor Nikolaos V. Sahinidis of Carnegie Mellon of Pittsburgh. Actually, the purpose is to optimize the superstructure above by means of the common software MATLAB®, then the successive step is to show how surrogate modeling works and to compare the new results with the original ones, especially for the easiest cases. In fact, it is fundamental to state if this technique is a valid strategy for the optimization of more complex heat exchanger networks.

2 FUNDAMENTALS

2.1 Overview on surrogate models

In general, engineers have to cope with experiments and simulations in order to describe the systems they are studying and from them they have to extrapolate their results and explanations. But these efforts often require long time, not only hours or days, but even weeks or months and, moreover, they are generally expensive. One way to reduce this big amount of work is generating surrogate models, that is, analytical functions which approximate as closely as possible the simulations previously obtained. Their target is to thoroughly describe data sets, so that any subsequent analysis is computationally cheaper and less time-consuming to perform. This type of models is quite common in engineering design and it is constructed on some data properly selected using well-known sampling methods, such as Latin hypercube or cluster sampling: that's why a data-driven approach is exploited. Every surrogate model comes out from a black-box modeling, since in the end inputs and outputs only can be viewed, while the knowledge of its internal working is not permitted. However, in the literature the variety of available approaches is quite wide, so the correct choice of surrogate is often a difficult task. Modern software, such as ALAMO, is able to reach a good accuracy even starting from few simulation evaluations; this allows researchers to design and optimize, for example, unit operations, reactors and processes etc., without spending too much money and time during the preliminary phase of data collection and data simulation. Therefore, the idea of using a simpler surrogate model to represent a complex phenomenon has more and more attracted the attention of a lot of people over the past three decades.

Mathematically, the surrogate modeling may be represented by the deterministic function $f: R^d \rightarrow R$, where d is the number of dimensions; it needs an input vector $X = (x1, x2, \dots, x^d)$, whose upper and lower bounds $X^L \leq X \leq X^U$ are known, and it generates a single output y. There may also be some constraints $f_j \leq 0$, $j \epsilon J$, where J is the set of all constraints. As it normally happens in engineering disciplines, it is assumed that the evaluation of the function and constraints is computationally expensive and their symbolic form is unknown; so, the analytical form of their derivatives is not available too. In order to overcome these difficulties, surrogate modeling produces a new function $\hat{f}(X)$ which approximates the initial one f. Consequently, it is worth to derive information about the objective function f directly from the surrogate $\hat{f}$, since its analytical form is known and it is cheaper to evaluate. All these properties make this technique of mathematical representation of the reality extremely useful in several applications, such as vehicle design, simulations, process modeling, process control, optimization, etc. In particular, in this context of guiding a search towards an optimum, a surrogate can be classified as local, if it is updated within an

iterative framework, and global, if it is fitted just once at the beginning and then used forever thereafter. Globally, problems requiring surrogates are divided into three classes:

1. First class: prediction and modeling;
2. Second class: derivative-free optimization (when the objective function to be optimized is expensive and its derivative is unavailable);
3. Third class: feasibility analysis (that is the objective is to satisfy design constraints too).

Probably, the first class is the main use of surrogates. Moreover, it is noteworthy to know that each class use different surrogate types. It is also important to have a selection methodology for regressions, which allows to choose the best model among a given set of models. In fact, sometime there can be overfitting, which occurs when there are too many input variables that might contain redundant information and, as a result, the surrogate doesn't generalize well on the test set. In order to face this problem, it is important to select most relevant variables in order to build simple yet effective surrogate models.

Before explaining in detail all the different types of surrogate models, it's important to differentiate between interpolation models, that are basically unbiased predictions at the sample data, and regression models, that are built by exploiting the minimization of the error between given data and model prediction under certain criterion.

2.2 Surrogate models

2.2.1 Linear regression

This is a very common approach, which shows the surrogate model composed of a linear combination of input variables:

$$\hat{f}(x) = w_0 + \sum_{j=1}^{d} x_j w_j$$

where x is the vector of inputs, d is the number of variables, that is the dimension of x, while w is the vector of length $d + 1$, which is obtained by minimizing the sum of squared errors between the actual data and the surrogate predicted value. Its form is the one of an unconstrained problem:

$$\min_{w} \|Xw - y\|_2^2$$

in which X is a matrix of size n by $d + 1$, where n is the number of sample points; all the elements in the first column of X are 1, while the ones of the successive columns correspond to the input vector. Instead, y is another vector of size n, which contains the function values at sample points. The analytical form of w is:

$$w = (X^T X)^{-1} X^T y$$

As one or more of the independent variables are perfectly correlated, the matrix $X^T X$ approaches the unitary value, so that coefficients w cannot be uniquely defined. This is typical of high dimensional problems, whose number of variables overcomes the quantity of data, and may be solved by reducing the variables with screening or regularization techniques. In effect, as the number of variables d in a problem increases for whatever reason, the bias on the data points decreases, but the variance becomes more and more significant, making a good prediction difficult: these are the effects of the aforesaid overfitting. So, those exceeding, unnecessary variables may be either removed with a subset selection or suppressed with a regularization.

Subset selection

It is a sort of trade-off between prediction error and regression model complexity while selecting a subset of variables, followed by the determination of the regression coefficients. There are some different methods to accomplish that target:

- The exhaustive search methods explore all possible subsets of features and select that one with minimum prediction error. They guarantee that the chosen model is the best fitting one, but computational complexity may be significant, as it increases a lot with number of subsets.
- The heuristic methods exploit some greedy approaches. One is the forward-stepwise regression, which starts from an empty set of variables and then it sequentially adds the ones that best fit the model. The quality of those new fitting variables is usually measured by an F-statistic, which uses sum squared error. Another approach is the backward-stepwise regression, that acts in the opposite way as it starts from including all possible variables and then it sequentially removes the ones with the minimum impact on the fit. The last approach is the forward-stagewise regression, which is similar to the first one, but in this case just the coefficient of the newly added variable is changed, while keeping the others fixed.
- The integer programming methods have the form of an optimization problem, in which an error measure EM is made of minimized subject constraints, that assure the desired subset selection by imposing an upper bound on the number of nonzero entries. Moreover, when it is needed, these approaches can ensure statistical properties, such as robustness and sparsity of the model. Anyway, some additional information is necessary a priori to make these methods effective, that is the prespecified number of variables, which is likely not available. An example of the formulation is reported here:

  $$\min EM$$
  $$s.t.$$
  $$\sum_{i=1}^{d} z_i = k$$
  $$w_{lL} z_l \leq w_l \leq w_{lL} z_l, l = 1, \dots, d$$
  $$z_l \in \{0,1\}, l = 1, \dots, d$$

  where z_l is a binary number for selection of the variables l, k refers to the number of subsets to be selected, $w_l$is a coefficient, $w_{lL}$and w_{lU} represent its lower and upper limits.

- The model fitness methods try to prespecify the number of selected variables by including penalties for nonzero terms, giving a trade-off between model complexity and prediction accuracy. In literature there are several fitness measures, such as:
  - Mean absolute error (MAE);
  - Mean squared error (MSE);
  - Akaike information criterion (AIC), that minimizes the discrepancy between the original distribution of data and the new one generated by the regression model by exploiting a discrepancy parameter (AICc represents a correction factor added to AIC for finite sample sizes);
  - Hannan-Quinn information criterion (HQIC);
  - Bayesian information criterion (BICc), whose objective is to minimize the approximate posterior probability;
  - Risk inflation criterion (RIC);
  - Mallow's Cp, which minimizes prediction error where mean squared error is the error measure.
- The Bayesian approach applies the probability theory assuming the unknowns as random variables. The probability distribution represents the uncertainty over those unknowns before obtaining samples when it refers to prior distribution, but it also represents the uncertainty after obtaining samples when it refers to posterior distribution. If M models are considered so that the i^{th} model is θ_m, which basically consists in a subset of variables, then the target is to select the model with highest posterior probability:

$$\Pr(\hat{f}_m|X) = \frac{\Pr(X|\hat{f}_m)Pr(\hat{f}_m)}{\sum_{kM}\Pr(X|\hat{f}_m X)\,Pr(\hat{f}_k)}$$

with

$$\Pr(\hat{f}_m|X) = \frac{\Pr(X|\hat{f}_m)Pr(\hat{f}_m)}{\sum_{kM}\Pr(X|\hat{f}_m X)\,Pr(\hat{f}_k)}$$

where X is the sample data set, while θ_k represents the unknowns in the surrogate model $\hat{f}_k$.

- The sure independence screening (SIS) method relies on learning sets of input variables according to their correlation with output ones y. The columns matrix X correspond each one to an input variable and they are all standardized, so that the resulting vector $X^T y$ contains those marginal correlations input-output and the least influent input variables can be filtered out.

- In the least angles regression method the parameters are added similarly to the forward-stepwise regression, but continuously monitoring the correlation input-output and new variables are sequentially added.

A table showing the expression of those fitness model measure is Table 1:

Table 1. Common model fitness measures. *(Sahinidis, 2017)*

$$\mathrm{AIC}_c = N \quad \log\left(\frac{1}{N}\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2\right) + 2p + \frac{2p(p+1)}{N-p-1}$$

$$\mathrm{HQIC} = N \quad \log\left(\frac{1}{N}\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2\right) + 2p\log(\log(N))$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2}{N-p-1}$$

$$\mathrm{C}_P = \frac{\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2}{\hat{\sigma}^2} + 2p - N$$

$$\mathrm{BIC} = \frac{\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2}{\hat{\sigma}^2} + p \quad \log(N)$$

$$\mathrm{RIC} = \frac{\sum_{i=1}^{N}(z_i - \mathbf{X}_i\beta)^2}{\hat{\sigma}^2} + 2p \quad \log(k)$$

Where $p < k$ is the number of coefficients, N is the number of sampled points and σ^2 is the error variance.

Regularization

Since in regression modeling certain variables may be accepted or discarded, the variance stays quite high and the prediction error is not decreased. That's why regularization may be implemented, causing a continuous reduction of regression model coefficients:

$$\min_w \|Xw - y\|_2^2 + C\|w\|_q$$

with

$$\|w\|_q = \left(\sum_{i=1}^{d} w_i^q\right)^{1/q}$$

where C is the parameter of magnitude of regularization that tries to penalize the regression coefficients w, which is the argument of the q^{th} norm. As a matter of fact, this norm has a significant effect on the properties of the regression model. Generally, two variants of regularization are used,

that is ridge regression and lasso regression, where the second one is able to set parameters exactly to 0. If the value of q is between 1 and 2, a mix of the properties of those two variants are used.
An alternative approach is the elastic-net regression, in which there is a linear combination of lasso and ridge penalty:

$$\min_{w} \|Xw - y\|_2^2 + C \sum_{j=1}^{d} (\alpha w^2 + (1-\alpha)|w|)$$

where α is a tuning parameter.
Otherwise, another approach is the adaptive lasso, which uses a weighted summation of coefficients as penalty term.
Finally, other approaches use a non-negative garrotte estimator, obtained by scaling coefficients of least squared regression.

2.2.2 Support vector regression (SVR)

It is basically a weighted sum of basis functions added to a constant term, whose general form is:

$$\hat{f}(X) = \mu + \sum_{i=1}^{n} w^i \psi(X, X^i)$$

It may be assumed that a simple basis function is $\psi(.) = X$, thus the surrogate becomes:

$$\hat{f}(X) = \mu + w^T X$$

The unknowns μ and w in the model are evaluated by the following mathematical optimization problem, that is represented by a minimization of combined contribution of model complexity and penalized outliers:

$$min \frac{1}{2}|w|^2 + \mathrm{C}\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\xi^{+(\mathrm{i})} + \xi^{-(\mathrm{i})}\right)$$

$s.t.$

$w.x^i + \mu - y^i \leq \varepsilon + \xi^{-(i)}$

$y\hat{}i - w.x\hat{}i - \mu \leq \varepsilon + \xi\hat{}(+(i))$

$\xi^{+(i)}, \xi^{-(i)} \geq 0$

where $\pm\varepsilon$ are the permitted deviations of sample points from the sampled points that generate a ε intensive tube, while $\xi^{+(i)}$ and $\xi^{-(i)}$ are slack variables that ensure the feasibility of the problem by

allowing outliers that do not fall within the ε intensive tube. A pre-defined constant $C \geq 0$ penalizes outliers. If the basis functions that are used are not linear, some additional hyper-parameters must be determined for each one. The SVR gives fast and accurate predictions, but in the end the time required to build this model is high, because a quadratic programming problem is necessary to estimate the unknown parameters.

2.2.3 Radial basis functions (RBF)

If n distinct sample points are given, the general formulation of RBF is:

$$\hat{f}(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|_2) + p(x)$$

Where $\lambda i \, \epsilon \, R$ are the weights that must be determined, $\|.\|$is the Euclidean norm and, finally, $\phi(.)$ is the basis function, whose most common forms are shown in the following table:

Type	Function $\phi(.)$
Thin plate spline	$\phi(r) = r^2 \log(r)$
Multi-quadratic	$\phi(r) = \sqrt{r^2 + \gamma^2}$
Gaussian	$\phi(r) = e^{-r^2}$

Table 2*. Commonly used radial basis functions. (Atharv Bhosekar, 2017)*

In particular, in the case of multi-quadratic and Gaussian basis functions, $r \geq 0$ and γ is a positive constant. By experience, none of the RBFs above is better than the others, nevertheless it is noted that putting together a cubic basis function and a linear tail, like the expression presented below, may be successful:

$$\hat{f}(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|_2) + a^T x + b$$

where ϕ is a n by n matrix with $\phi_{ij} = \phi(\|x - xi\|_2)$.

The parameters a, b and the weights λ are evaluated by solving a system of equation like this:

$$\begin{pmatrix} \phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

with

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{pmatrix};\ \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}; c = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \\ a \end{pmatrix}; F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}$$

An extension may be implemented to the RBF: the bumpiness function, which is explained successively.

2.2.4 Kriging

This type of surrogate model is a sort of realization of a stochastic process and it is also known as Gaussian process regression for machine learning. It was first proposed in the '50s in the field of geostatistics, then it was extended to computer analysis. The general formulation is given by:

$$\hat{f}(x) = \sum_{j=1}^{m} \beta_j f_j(x) + \varepsilon(x)$$

where $f_j(x)$ the known independent basis functions, m is their total number and x is the location around which those basis functions define the trend of mean prediction. Then, β_j are unknown coefficients and $\varepsilon(x)$ is a random error around location x with a normal distribution and with a mean distribution equal to zero. This last term permits the modeler to have an assessment of uncertainty in addition to the predicted value and this an important peculiarity typical of the Kriging regression. Furthermore, there is a kriging predictor with the form:

$$\hat{f}(x) = f(x)^T \beta^* + r(x)^T \gamma^*$$

where $f(x) = [f_1(x), \dots, f_m(x)]^T$, β^* is the vector of generalized least squared estimates of $\beta = [\beta_1, \dots, \beta_m]^T$ and the column vector $r(x)$ correlates $\varepsilon(x_i)$ to $\varepsilon(x_j)$.

β^* and γ^* are evaluated as:

$$\beta^* = (F^T R^{-1} F)^{-1} F^T R^{-1} y$$

$$\gamma^* = R^{-1}(y - F\beta^*)$$

where y are observations at available data, R is the covariance $n \times n$ matrix in which every element links $\varepsilon(x)$ to ε(xj), while $F = \left[f\left(x^{(1)}\right), \ldots, f(x^{(n)})\right]^T$ is an $n \times m$ matrix. The variance of the problem is estimated as:

$$s^2(x) = \hat{\sigma}^2[1 - r^T R^{-1} r]$$

with

$$\hat{\sigma}^2 = \frac{1}{n}(y - F^T\beta^*)^T R^{-1}(y - F^T\beta)$$

It is remarkable that some hyper-parameters, that is a set of unknowns, are used to calculate r and R and they are evaluated by means of maximum likelihood ML, usually provided in logarithmic form:

$$\log ML(\theta) = -\frac{1}{2}[n \ln(2\pi\sigma^2) + \ln det(R(\theta)) + (y - F\beta^*)^T R(\theta)^{-1}(y - F\beta^*)/\sigma^2]$$

Correlation models

Anyway, the Kriging surrogate model expression is composed of 2 parts: the first is the regression model, the latter is the correlation one.

Normally, the random variables $\varepsilon(x)$ are assumed to be correlated and, for deterministic and continuous functions, if two samples are close to each other, then their predicted values are close as well. So, there is an elevated correlation among random variables which may decrease as two distinct samples get further. In Table 3 the most common correlation models are reported:

***Table 3**. Commonly used correlation models in Kriging surrogates. (Atharv Bhosekar, 2017)*

Commonly used correlation models in Kriging surrogates.

Name	Correlation model
Exponential	$\exp(-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert^{p_j}), 0 < p_j < 2$
Squared exponential	$\exp(-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert^{2})$
Linear	$\max\left\{0, 1-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert\right\}$
Spherical	$1-1.5\xi_j+0.5\xi_j^3, \xi_j = \min\left\{1, \sum_{j=1}^{d}\theta_j\lvert m_j\rvert\right\}$
Matern	$\prod_{j=1}^{d}\frac{1}{(\nu_j)2^{\nu_j-1}}\left(\theta_j\lvert m_j\rvert\right)^{\nu_j}K_{\nu_j}(\theta_j\lvert m_j\rvert)$

in which m_j represents the distance between two points, θ_j and p_j are hyper-parameters and d is the number of dimensions of the original problem. In particular, inside Matern correlation model, Γ is the Gamma function, K_{ν_j} is the modified Bessel function of order ν_j and $\nu_j > 0$ is a coefficient controlling the differentiability of correlation model with respect to input variable x_j.

Variants of Kriging regression models

Still, there are several variants of Kriging and everyone is essentially based on a mean prediction model $f(x)^T\beta$:

1. Simple Kriging, where $f(x)^T\beta$ is assumed to be constant and known;
2. Ordinary Kriging, which considers $f(x)^T\beta$ constant, but unknown;
3. Universal kriging, also called Kriging with a trend, regards $f(x)$ as any other prespecified function of x, usually in the form of a lower order polynomial regression.

However, having a specified $f(x)$ right at the beginning may lead to inaccuracy, which can be avoided some variable selection techniques available in literature. For example, in the blind Kriging the unknown trend is given by a Bayesian approach and it tries to have a model with maximum posterior probability. Another technique is the generalized degrees of freedom, where estimator of mean squared error is minimized. Other strategies for variable selection utilized in Kriging involve penalized likelihood functions, that is by adding a penalty term. Moreover, there are different combinations of mean prediction terms $f(x)$ and of correlation models for random error $\varepsilon(x)$, thus obtaining with multiple Kriging models. Nevertheless, having complex regression terms may result in lower accuracy and extra computational expense.

Nugget effect

Kriging is an exact interpolation technique, which means that the predicted value matches exactly the underlying black-box function at the initial sample points. This may lead to high oscillations of the prediction, but this phenomenon can be deadened by adding a nugget term ε to the covariance matrix I inside the regression, thus attaining the following equation:

$$R^{mod} = R + \varepsilon I$$

As the distance between two points approaches zero, the correlation gets further and further from 1 and a singular, or ill-conditioned, covariance matrix is considered. Hence, when two sample points are too close to each other or hyper-parameters in the covariance model are too near to zero, the nugget effect allows the covariance matrix to maintain its conditioning.

Computational aspects of Kriging

There are some characteristics of Kriging that are worth to understand in order to better apply this method.

First, it involves an inversion of covariance matrix, which may be computationally expensive with high number of samples. Second, the Kriging maximum likelihood estimator ML is optimized and, in addition, is dependent on the inverse covariance and highly non-convex, thus implying multiple evaluations to search for global optima. Actually, getting stuck at a local optimum could affect Kriging surrogate prediction, but the exploitation of simple covariance equations can allow to overcome this issue. Anyway, as involved dimensions increase, the aforesaid non-convexity and computational efforts become a trouble. In particular, the number of hyper-parameters of correlation models depends on dimensionality of the problem, but it can be reduced by using partial least squares, so that it is possible to find a solution more efficiently up to 100 dimensions. Otherwise, cross validation may be a robust alternative, even though the variance may be quite high. Still, when the number of data points is extremely huge, there are some other approaches:

- By dividing the covariance matrix into small matrices of size r, which is the number of implemented basis functions;
- By tapering the covariance, resulting in a covariance matrix, where most of insignificant elements are set to zero, so that its inversion becomes easier;
- By choosing just a subset of data points to generate Kriging surrogate model;
- By combining the approaches here above.

By the way, the experience reveals that if a covariance matrix is positively definite, the computational efforts of its inversion may be significantly reduced.

2.2.5 Mixture of surrogates

Most of times, none of all surrogate model type is able to properly perform for all type of problems and, at the same time, it is not always possible to implement multiple choices of them to select the best one. So, it is often useful to utilize a combination of surrogates, whose general form is:

$$\hat{f}(x) = \sum_{i=1}^{n} w_i(x)\hat{f}_i(x)$$

where $w_i(x)$ is the weight relative to the i^{th} surrogate of design point x and the summation of weights $\sum_{i=1}^{N} w_i$ is set to 1. Therefore, if all surrogates $f_i(x)$ are equal, then the weighted mixture will predict the same value. Those weights may be estimated with different approaches, such as a global cross validation called PRESS or by assigning probability to the surrogates with the help of an error metric. As a result, this strategy of multiple surrogates gives more flexibility, since it emphasizes more on good surrogate despite of bad one.

2.3 Derivative-free optimization (DFO) and surrogates

The DFO occurs when function derivative information is not symbolically or numerically available and they are classified as:

A. Local search algorithms, called local DFO, reach a local optimum or refine the solution by starting from an initial guess;
B. Global search algorithms, called global DFO, use also a component which can make the research depart from a local minimum.

More in details, local DFO may be divided into two subsets:

- Direct search algorithms, that do not rely on surrogates and sequentially examine the points generated by a strategy, sometimes identifying geometric patterns. This is the major class of local DFO and most common ones are the Hooke and Jeeve's algorithm or the simplex method.
- Model-based algorithm, that take into account surrogates.

Instead, global DFO essentially comprises algorithms that do not use surrogate models and, in particular, their approach includes a partition of the feasible space or a stochastic study. An example is the DIviding RECTangles (DIRECT) algorithm (Jones D. R., 1993).

2.3.1 Model-based local DFO

These techniques are called trust-region methods too and implement surrogate models in a neighborhood, also called trust region, of a given sample location. They should precise inside that studied area, whose size depends on the radius related to the desired accuracy. This method stops as the trust region becomes small enough. Several types of surrogate were developed for this technique and the most common ones are linear interpolation models implemented on COBYLA (Constrained Optimization BY Linear Approximation, (D., 1994)) algorithm or quadratic ones. While the first gives problem on curves, the latter is generally more suitable and can be formulated as:

$$\hat{f}(x_k + s) = f(x_k) + s^T g_k + \frac{1}{2} s^T H_k s$$

where k is the iteration index, x_k is the current iterate value, g_k belongs to the domain $\boldsymbol{R}^d$ and H_k is a $d \times d$ matrix. $\frac{(d+1)(d+2)}{2}$ is the number of sample points which allow to determine g_k and H_k and it obviously increases with d: for example, for 30 dimensions this number is around 500. So, to curb this problem there are some undetermined quadratic interpolation models, whose task is to reach a stationary local optimum, so that they converge locally. Alternatively, some people use interpolation models with cubic basis functions and a linear tail, that are also proven to be globally convergent. In

addition, some modifications may be brought to the RBF models, obtaining the Optimization by RBF Interpolation in Trust-regions (ORBIT) algorithm, which was later extended to tackle constrained problems too (Wild S. M., 2008). Finally, a similar strategy exploits Kriging and it is called Efficient Global Optimization (EGO).

2.3.2 Model-based global DFO

With the help of these algorithms, non-convex surrogates can be optimized and exploited. In fact, the generated surrogate model is related to the whole feasible space or to more parts of it, then a proper function guides to all local and global optima by means of a balance between local and global search. This function can be the Expected Improvement (EI) which is maximized or bumpiness which is minimized: both of them will be studied later in this thesis work. There is another approach for global search where RBF functions are sequentially optimized and updated over the feasible space. Alternatively, multiple local search may be simultaneously driven starting from different points and with a certain strategy: if the algorithm gets stuck in a local minimum, it restarts from a new sample point. ARGONAUT, that is Algorithms for Global Optimization of constrained grey-box computational problems, is a surrogate modeling software which contains all the surrogates analyzed till now (Boukouvala F., 2016).

2.4 Feasibility analysis and surrogates

Feasibility is defined as the ability of process to satisfy all relevant constraints, while feasibility analysis is the identification of the conditions under which the process can be feasible. In chemical engineering, common examples related to feasibility analysis are all the constraints over processes, such as product demand, environmental conditions, safety or material properties. Other applications of the concept of feasibility are the profit maximization or the required specifications of a material or a product. A feasibility function is formulated here below:

$$\psi(d,\theta) = \min_{z} \max_{j\epsilon J}\{f_j(d,z,\theta)\}$$

where d is design variable and z is the control one, while the bounds on z are $z\,\epsilon\, Z = \{z: z^L \leq z \leq z^U\}$; then, θ represents uncertain parameters $\theta\,\epsilon\, T = \{\theta: \theta^L \leq \theta \leq \theta^U\}$; finally, $f_j(d,z,\theta)$ are the constraints. The functioning of this function is quite simple: checking if all constraints f_j can be satisfied for a given design variable d while adjusting the control variable z. Hence, if $\psi(d,\theta)$ is positive, it means that one or more constraints are violated, that is the design is unfeasible; instead, if $\psi(d,\theta)$ is equal to 0, then we are exactly on the boundary of the feasible region; in the end, if $\psi(d,\theta)$ is negative, the design is feasible under the conditions that were imposed.

In all process designs, the feasibility is strictly linked to the intrinsic uncertainty of input parameters θ. That's why it is quite important to define the concept of flexibility, that is the ability of a process to remain feasible even though some nominal deviations of uncertain parameters act on it. This property is measured by the flexibility test problem, expressed in general as a max-min-max form:

$$\chi(d) = \max_{\theta} \min_{z} \max_{j\epsilon J}\{f_j(d,z,\theta)\}$$

In practice it checks if feasibility function $\psi(d,\theta)$ is non-positive over the entire range of uncertain parameters θ. In literature there are several approaches which apply feasibility function and flexibility in different ways. One is the quantification of the feasible region by inscribing it into the largest hyper-rectangle. In other approaches the feasibility function is assumed to be known, but they need big computational efforts when feasibility is to be evaluated, so they cannot be often used. Therefore, they are frequently substituted by black-box approaches, making surrogate-based methods a promising alternative.

The key-idea is to approximate that feasibility function through the development of a surrogate after providing input parameters and black-box simulation outputs. The main factors to be considered are

the choice of surrogate model and the sampling strategy and there are several approaches addressing both, such as High Dimensional Model Representation (HDMR), Kriging or RBF. In addition, Convex Region Surrogate (CRS) is useful to represent a nonlinear and nonconvex feasible space by combining more convex regions. Otherwise, another approach exploits random line search for detecting boundary points of feasible region.

Quantity and quality of sampling sets strongly affect the global quality of the final surrogate model. Indeed, increasing sample size will generally lead to a better prediction, but the sampling will be costlier. In the case of feasibility study, the presence of constraints makes sampling requirement higher than the case of single objective prediction and this is the reason why adaptive sampling techniques are implemented. To tackle this issue, Kriging is proposed as well as a modified version of EI function, whose expression is shown here below:

$$\max_{x} EI_{feas}(x) = \mathrm{s}\phi\left(-\frac{\mathrm{y}}{\mathrm{x}}\right) = \mathrm{s}\left(\frac{1}{\sqrt{2}}e^{-0.5\left(\frac{y^2}{s^2}\right)}\right)$$

where $EI_{feas}(x)$ is the function to be maximized at the value x, y is the surrogate model predictor, s is the standard error and $\varphi(.)$ is the normal probability distribution function.

In feasibility analysis the target is to find a surface which defines the boundary of the feasible space within the box bounded design space, while global optimization searches for a single optimum. In order to address this problem, both Kriging and RBF surrogates may be used, resulting in similar accuracies. The first multiplies feasibility function values x of nearby samples and check if the product is positive, meaning that they are at the same side of feasible boundary; basically, it uses the Kriging variance to ensure the exploration and to better define boundaries. The latter uses bumpiness measure to attain prediction error, it substitutes that in max EI_{feas} function, ended up with new sample points for evaluation.

2.5 Sampling

Sampling is defined as the process of generating data points that allow to build surrogate models, whose performance strongly depends on quality and quantity of samples. Hence, efficient sampling strategies are on great interest because they allow modelers to maintain the quality of surrogates while reducing excessive sampling efforts.

The stationary sampling method relies in geometry or pattern, such as grids, full and half factorial design or Box-Behnken design. A common example of stationary sampling is the Latin Hypercube Sampling (LHS), which is a stratified strategy with every sample drawn from each stratum (McKay, 1979). Another example is the quasi-random Sobol and Halton sampling, where samples are drawn through low-discrepancy sequences (Sobol, 1967). Instead, the adaptive sampling method sequentially selects new sample locations starting from a limited number of initial samples, which come from stationary sampling. The task is to minimize sampling requirements in order to obtain more, better-positioned samples which improve quality of surrogate. Moreover, there is a great trade-off between exploration and exploitation: basically, the first searches in the most unexplored region and is useful to escape from local optima; instead, the latter refines the region near existing samples for better understanding and helps the surrogate improve available optimum.

2.5.1 Expected improvement function (EI)

The Expected Improvement function EI is largely used in Kriging method for both exploration and exploitation and its general form at sample location x is:

$$EI(x) = \left(f_{min} - \hat{f}(x)\right)\boldsymbol{\Phi}\left(\frac{f_{min} - \hat{f}(x)}{s}\right) + s\phi\left(\frac{f_{min} - \hat{f}(x)}{s}\right)$$

where $\boldsymbol{\Phi}(.)$ represents the standard normal density function, while $\phi(.)$ acts as the probability distribution function; moreover, $\hat{f}$ is the surrogate model predictor, that is the predictor value, f_{min} is the current minimum function value and s is the standard deviation. $EI(x)$ gives lower values for decreasing $f(x)$ and increasing s, which means exploration and exploitation, respectively. As a matter of fact, the trade-off between these two operations is included into maximization of EI function, which may have multiple local optima causing numerical problems.

2.5.2 Bumpiness function

Instead, the bumpiness function is widely used for RBF surrogate, which works by solving the above-mentioned system of equation plus the minimization of the bumpiness function itself. The general expression is provided here:

$$\min g_n(y) = (-1)^{m_0+1}\mu_n(y)\left[\hat{f}(y) - f_n^*\right]^2 \ , y \in D\backslash\{x_1, x_2, \dots, x_n\}$$

where y is an unsampled point, f_n^* is the target value and $\mu_n(y)$ is the coefficient of the new term $\phi(\|x - y\|)_2$ in the surrogate $\hat{f}_n(x)$ as a new y is added. In addition, m_0 is a constant depending on the basis function (1 for cubic and thin plate splines, 0 for linear and multi-quadratic, -1 for Gaussian). Moreover, $\mu_n(y)$ is calculated as the nth element of vector v, that comes from the solution of the following system of equations:

$$\begin{pmatrix} \phi_y & P_y \\ P_y^T & 0 \end{pmatrix} v = \begin{pmatrix} 0_n \\ 1 \\ 0_{d+1} \end{pmatrix}$$

Its evaluation is computationally expensive, as an $o(n3)$ operation is necessary to find μ_n. Hence, by minimizing the bumpiness function, both exploration and exploitation are facilitated while depending on f_n^*. However, big negative values of f_n^* makes the search global and focused on exploration, whilst approaching the current optimal solution makes the search local and focused on exploitation.

2.5.3 Other approaches

The adaptive sampling can be considered as a DFO problem, where objective function is expressed as the difference between the true function and the surrogate one:

$$max\left(\frac{f(x) - \hat{f}(x)}{f(x)}\right)^2 \ , x_L \leq x \leq x_U$$

where $\hat{f}(x)$ is the surrogate, $f(x)$ is the true function, while x_L and x_U are the lower and upper bounds of the sample value x within which the error must be maximized. Some approaches rank and weigh exploration and exploitation basing on what is more convenient in the moment, by identifying two separate measures for each of them.

In particular, for the first technique, the sum of squares of the distance acts as new sample from all the previous samples; for the second one, a departure function quantifies the impact of new sample added near an already sampled location:

$$\Delta_j(x) = \hat{f}(x) - \hat{f}_j(x), j \epsilon S$$

where $\hat{f}(x)$ is the surrogate built on sampled sets S and $f_j(x)$ is the surrogate built on all points except the j^{th} one.

2.6 Surrogate model validation

The surrogate model validation allows the assessment of the reliability of the surrogate itself and it is a very important issue. In fact, an inaccurate surrogate model may probably result in bad optimization, false predictions and waste of resources. Hence, these validation techniques automatically select the best surrogate models among a lot of candidates and tune hyper-parameters and it is possible to compare predictions with true values when problems have just few dimensions. However, one big problem of surrogate modeling is that the same data set cannot be used both to build it and to validate it: that's why only the initial part of the available data, called training set, allows to create the model, while the remaining part allows to test its accuracy and it is called test set, whose error is quantified by some validation metrics. Their mathematical expressions are shown in the Table 4.

Table 4. Common surrogate validation metrics. *(Atharv Bhosekar, 2017)*

Validation metric	Formula
Explained variance score	$1 - \frac{Var\{y-\hat{y}\}}{Var\{y\}}$
Mean absolute error	$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \lvert y_i - \hat{y} \rvert$
Mean squared error	$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \left(y_i - \hat{y}\right)^2$
Median absolute error	$\text{median}\left(\lvert y_1 - \hat{y}_1 \rvert, \ldots, \lvert y_n - \hat{y}_n \rvert\right)$
R^2 score	$1 - \frac{\sum_{i=0}^{n_{samples}-1} y_i - \hat{y}_i^2}{\sum_{i=0}^{n_{samples}-1} y_i - \overline{y}^2}$
Relative average absolute error	$\frac{\sum_{i=1}^{n_{samples}} \lvert y_i - \hat{y}_i \rvert}{n_{samples} * STD}$
Relative maximum absolute error	$\frac{\max\left(\lvert y_1 - \hat{y}_1 \rvert, \lvert y_2 - \hat{y}_2 \rvert, \ldots, \lvert y_n - \hat{y}_n \rvert\right)}{STD}$

In those matrices, y, $\hat{y}$, $nsamples$, and $\bar{y}$ are true value, surrogate predicted value, number of samples and mean predicted value, respectively. Also, some authors use normalized mean squared error and average Negative Log estimated Predictive Density (NLPD) for heteroscedastic Gaussian process regression which penalizes over-confident as well as underconfident predictions. Other people implement the Mahalonobis error that avoids uncorrelated errors through full predictive covariance. Furthermore, the PREdiction Sum of Squares (PRESS) exploits the Root Mean Square Error (RMSE) to select the surrogate among multiple possible choices and it results to be more and more useful as number of sample points increases.

Here there is the expression of PRESS vector, which represents the vector of errors from carrying leave-one-out cross validation:

$$PRESS_{RMS} = \sqrt{\frac{1}{n_{samples}} \tilde{e}^T \tilde{e}}$$

Another measure of relative model performance which analyzes the results through test sets is the Error Factor (EF):

$$EF = \frac{RMSE}{RMSE_{best}}$$

where $RMSE_{best}$ is the minimum RMSE of the surrogate models under comparison for particular problems.

Some common approaches to implement the model validation are resampling strategies such as:

- Cross-validation, in which available data is divided into k blocks and there is equal number of data points in each one. Then, data from $(k-1)$ blocks are training sets and the remaining one is a test set. Afterwards, this process is repeated for all possible combinations of $(k-1)$ blocks and an appropriate validation metrics estimates the error on test set. There may be some troubles when available data are limited, as using part of data for model building is more difficult.
- In leave one out cross-validation, number of subsets k equals number of data points or observations, so that the surrogate is gradually built by leaving only one data point each time. in addition, a sampling set is considered inappropriate to build a good surrogate if removal of one data point strongly affects new model.
- Bootstrapping allows repeated samples in the training set and its size can even be equal to the size of actual data. Actually, the number of subsets k chosen for bootstrapping is usually much higher than that of other validation approaches.

In this work, ALAMO will be exploited to extrapolate a surrogate model of a heat exchanger network after a previous optimization on MATLAB® (MATrix LABoratory).

2.7 Heat exchange

In every industrial process the heat exchange is a fundamental transport phenomenon which helps it properly run: in particular, it can allow to achieve the desired temperatures in all the different units and fluxes, it can provide the necessary energy to a reaction or a designed system in order to make a transformation occur, it can subtract or recover energy to have a good optimization of the operating costs or it can play a crucial role in the safety control of a plant. These are just few, very common examples of application that heat exchange can have in chemistry, even if it's largely used in several other fields too.

Therefore, due to its priority importance, heat exchange needs to be well designed according to both the wanted specifications and physical limits of the considered system. When it's implemented in a plant, in reality it may assume a lot of different configurations, which may be conventional or not and may be similar or not to each other, but for sure each one will have its own characteristics. By the way, in most of cases heat exchange is achieved by means of specific pieces of equipment called heat exchangers.

2.7.1 Definition of heat exchanger

The heat exchanger is a typical unit operation working in lots of chemical plants, whose target is to transfer the heat between two or more fluids, and it can be used in both cooling and heating processes. In general, these fluids, even called working medium, may be not in contact, that is they may be separated by a solid, impermeable wall (often metallic), or they may be in direct contact one each other, obtaining their mixing. Nowadays, this energy exchange equipment is widely used in petrochemical industry, HVAC refrigeration, aerospace and so many other fields.

***Figure 2**. U-tube heat exchangers. (Boilers, 2015)*

***Figure 3**. U-tube heat exchangers. (Boilers, 2015)*

2.7.2 Classification according to flow arrangement

They may first be divided into three major classes according to their flow arrangement:

a) Parallel flow: the hot and cold fluids enter the unit at the same side, travel along it in parallel and leave together at the other end.

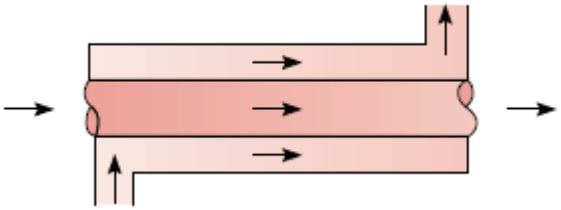


Figure 4. Concentric tube heat exchanger with parallel flow arrangement. *(Incropera, 2007)*

b) Counter flow: each fluid enters the unit from the opposite side with respect to the other one and flow in opposite directions.

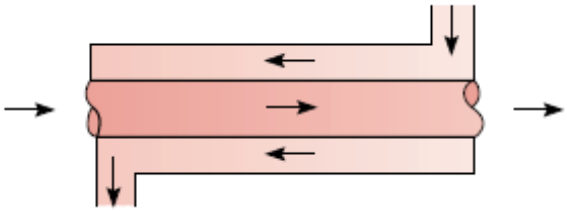


Figure 5. Concentric tube heat exchanger with counter flow arrangement. *(Incropera, 2007)*

c) Cross flow: the two fluids travel one perpendicularly to the other. Furthermore, this arrangement may be divided into two subsets: the finned and unfinned tubular heat exchangers. In the first configuration, the flux is said to be unmixed as some fins constrain the motion in the direction y perpendicular to the main-flow one x, causing a variation of the fluid temperature along x and y. In the second case, there are no fins, so mixing of fluid occurs and the temperature varies essentially in the main direction x. Therefore, for the finned heat exchanger both fluids are unmixed, while for the unfinned unit the shell-side fluid is totally mixed and the tube-side one is unmixed.

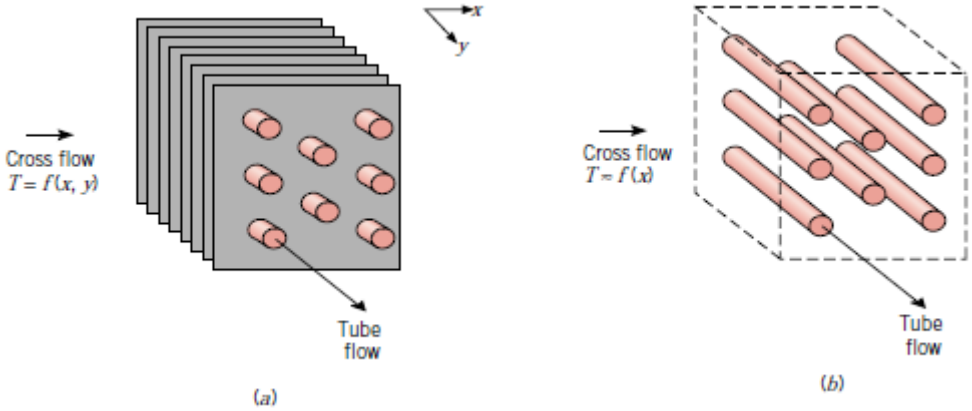


***Figure 6**. Cross-flow heat exchangers. (a) Finned with both fluids unmixed. (b) Unfinned with one fluid mixed and the other unmixed. (Incropera, 2007)*

From a general point of view, we can say that the counter-flow design is the most efficient, because the average temperature difference throughout the entire unit is higher than in the other two configurations, so this gives the highest heat transfer per unit mass.

The main index of the efficiency of a heat exchanger is related to an appropriate mean temperature. In simple systems this is the log mean temperature difference (LMTD):

$$\Delta T_{ml} = \frac{\Delta T_2 - \Delta T_1}{\ln \frac{\Delta T_2}{\Delta T_1}}$$

where ΔT_1 and ΔT_2 varies depending on the configuration of the heat exchanger, as it will be explained later.

Otherwise, when the LMTD is not available, there some other methods to approximate a temperature difference, like the NTU method, whose formulation is available in many engineers' handbooks (Incropera, 2007).

2.7.3 Types

It is possible to classify some main types of heat exchangers, essentially basing on their heat transfer pattern:

Double pipe: it's the simplest type and it's just composed of two metallic, contacting pipes; its design and maintenance is very cheap, but the maximum efficiency is quite low and the surface area requested in large-scale plants is huge. So, it's a good choice for small processes.

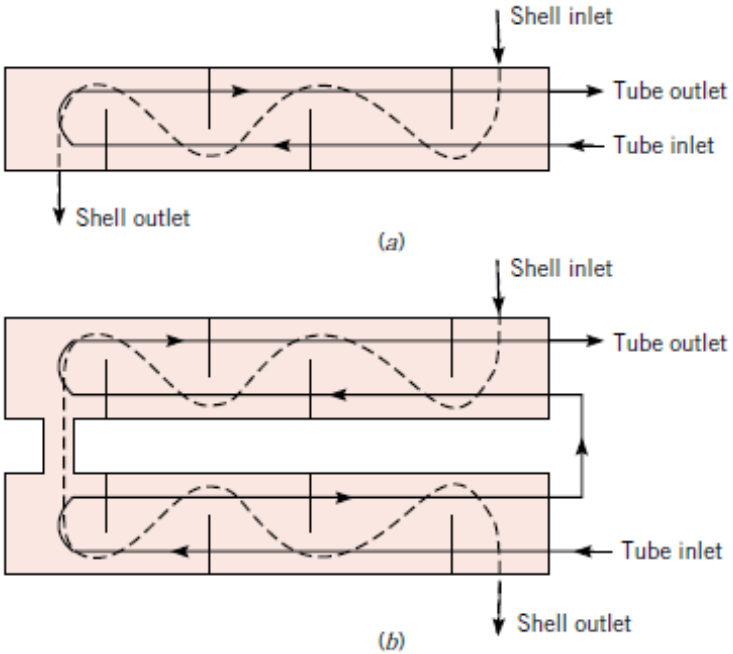


***Figure 7**. Shell and tube heat exchangers. (a) One shell pass and two tube passes. (b) Two shell passes and four tube passes (Incropera, 2007)*

Shell and tube: it is composed of a bundle of pipes inside a vessel, generally at high pressure. One of the two fluids runs in the tubes (called tube side) and the other one flows in the vessel (called shell side) by completely enwrapping the tubes, so that the heat exchange between those two fluids occurs through their wall, which is very commonly metallic. In effect, the starting temperatures in both sides are different, thus there is the requested driving force of the heat flux. The fluids may be either in liquid or gaseous and this heat exchange can stand both single-phase and double-phase systems. It is very often installed in oil refineries and large chemical processes. Finally, the flow arrangements that this type of unit can support are all the main ones previously analyzed (Incropera, 2007).

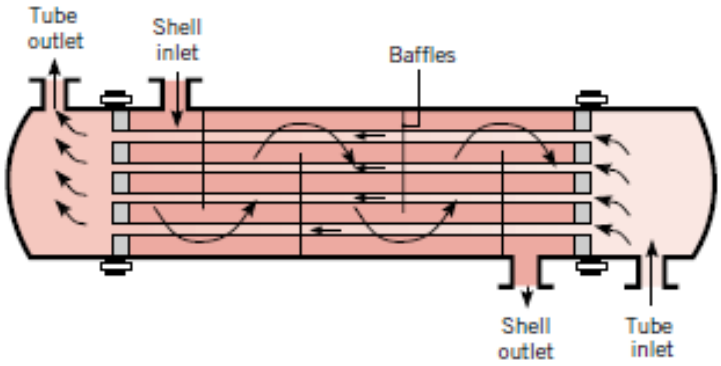


***Figure 8**. Shell-and-tube heat exchanger with one shell pass and one tube pass (the simplest one) with cross-counterflow flow arrangement as internal mode of operation. (Incropera, 2007)*

Plate: it uses metal pates to transfer heat between two fluids, so that they are spread out over those plates and the exchange surface area is significantly larger than conventional heat exchangers. This characteristic enhances the transfer and increases the speed of temperature change. The plate design is specialized in medium- and low- pressure fluids, while welded, semi-welded and brazed heat exchangers are applied to high-pressure cases or when the required product is quite compact. In general, the plate heat exchanger is composed of multiple, very thin chambers that are all alternatively separated at their largest surface by corrugated metal plates. Sometimes, these plates are assembled with tubes whose diameter may have different forms, such as flat tubes or circular tubes, so that one fluid passes through the tubes and the other fluid flows inside the plate side. The most used material is stainless steel due to its strength and resistance to high temperature and corrosion. In addition, there are some rubber sealing gaskets around the edge of the plates, which have some previously designed troughs to guide the flow and maintain its turbulence. All the plates are compressed together in order to form a rigid frame arranged in parallel flow channels with alternating hot and cold fluids and their configuration can be usually simplified into a manifold system with two manifold headers: one for dividing fluids (U-type) and one for combining them (Z-type). The minimum temperature difference of approach is approximatively 1°C for the plate heat exchanger, while it is about 5°C for the shell and tube type, so the heat transfer phenomenon is more efficient. Furthermore, for the same amount of heat exchanged, the size of the plate heat exchanger is smaller thanks to the intrinsic distribution of the exchange area, which can be incremented or decreased by simply adding or removing plates from the stack (Incropera, 2007).

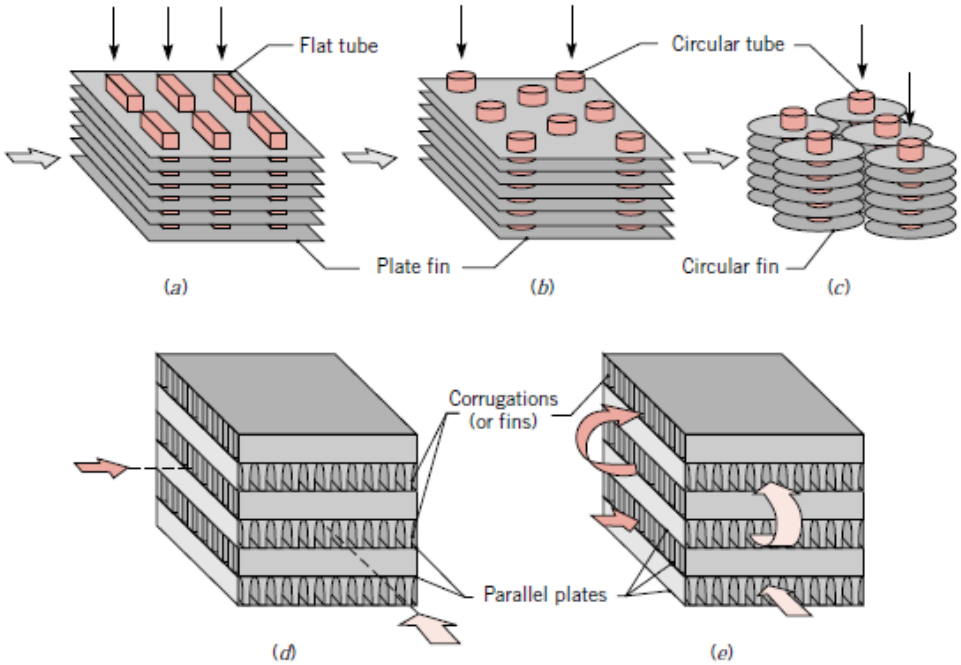


***Figure 9**. Plate heat exchanger configurations. (a) Fin–tube (flat tubes, continuous plate fins). (b) Fin–tube (circular tubes, continuous plate fins). (c) Fin–tube (circular tubes, circular fins). (d) Plate–fin (single pass). (e) Plate–fin (multipass). (Incropera, 2007)*

Direct contact: it involves heat transfer between hot and cold streams of two phases without any separating wall and it is classified as gas-liquid, immiscible liquid-liquid, solid-liquid or solid-gas. The first one is the most used, especially for air conditioning, humidification, condensing units and industrial hot water heating; in particular, the liquid phase interacts with the gas in the form of drops, films or sprays (Incropera, 2007).

***Figure 10**. Real heat exchanger. (Industry, 2019)*

2.7.4 Problems and maintenance of heat exchangers

In order to monitor integrity and cleanliness of the heat exchange, the overall heat transfer coefficient is periodically estimated from exchanger flow rates and temperatures, as it tends to decline over time due to some problems, like fouling. This phenomenon is the most common problem of heat exchangers and it is essentially the progressive deposition of impurities on heat exchange surface. Indeed, it makes heat transfer effectiveness significantly decrease and there a lot of different dynamics, such as precipitation of dissolved compounds or caused by some type of reactions, low wall shear stress, too low or too high fluid velocity. Obviously, the rate of fouling is determined by the rate of particle deposition less re-entrainment/suppression. One of the main causes is crude oil heating, as it may contain insoluble asphaltenes which fall down and gradually close the cross section. Also, cooling water typically brings dissolved solids and suspended colloidal solids, which can precipitate on exchange wall because of high local temperature or low flow velocities, less than 3 [ft/s]. In order to explain the fouling, it may be useful to go deeper into detail. For example, we can consider a double-pipe exchanger, in which the hot fluid flows in the inner pipe, like the one shown in the figure below.

Figure 11. *Double-pipe heat exchanger with counter-current flow arrangement with hot fluid flowing in the inner pipe. (Serth, 2007)*

The heat flux may be expressed as:

$$q = UA\Delta T$$

Where the driving force for the heat transfer is $\Delta T = \Delta T_{ml}$, that is the logarithmic temperature difference, which will be better described later in this work. The thermal resistance may be defined as:

$$R_{th} = \frac{1}{UA}$$

But this quantity is actually composed of three resistance in series, namely, the convective resistance between hot fluid and pipe wall, the conductive resistance of pipe wall and the convective resistance between pipe wall and cold fluid; so, they may be generically expressed as:

$$\frac{1}{UA_o} = \frac{1}{h_i A_i} + \frac{\ln(D_o/D_i)}{2\pi kL} + \frac{1}{h_o A_o}$$

where k is the thermal conductivity, D_i and h_i are the diameter and the heat transfer coefficients for the flow relative to the inner pipe, while h_o and D_o are the same two parameters relative to the outer pipe. The formulas of internal area A_i and external area A_o are:

$$A_o = \pi D_o L$$
$$A_i = \pi D_i L$$

The previous equation can by multiplied by A_o and the inverted, obtaining:

$$U = \left[\frac{D_o}{h_i D_i} + \frac{D_o \ln(D_o/D_i)}{2\pi kL} + \frac{1}{h_o}\right]^{-1}$$

However, this equation is correct only if the heat exchanger and all its surfaces are perfectly clean. In fact, as said above, most fluids leave impurities because of fouling and a film of dirt or scale builds up on the walls as time runs, resulting in decreased performance of the heat exchanger due to those added thermal resistances. In order to account for this phenomenon, some fouling factors R_{Di} and R_{Do} have been empirically determined and they are available in literature; they represent the thermal resistances of the dirt films on the inside and the outside of the inner pipe, respectively, multiplied by the corresponding surface areas, as shown below:

$$R_{Di} = R_{Dth} A_i \longrightarrow R^i_{Dth} = \frac{R_{Di}}{A_i}$$
$$R_{Do} = R_{Dth} A_o \longrightarrow R^o_{Dth} = \frac{R_{Do}}{A_o}$$

Adding these two contributions to the aforesaid expression of the overall heat transfer coefficient U (remembering to multiply both for A_o) yields:

$$U_D = \left[\frac{D_o}{h_i D_i} + \frac{D_o \ln(D_o/D_i)}{2\pi kL} + \frac{1}{h_o} + \frac{R_{Di} D_o}{D_i} + R_{Do}\right]^{-1}$$

where U_D expresses the overall coefficient after fouling has occurred. It is clear that the fouling factors have the effect to decrease the final value of U_D, which increases the heat transfer area. Of course, U_D gives more realistic results and fouling factors should be chosen so that heat exchanger has a reasonable operating period before starting cleaning procedure, which is generally quite expensive and needs to shut down the whole unit. In addition, fouling factors R_{Di} and R_{Do} can act as safety factors in the design. Anyway, the global effect on the entire system is that larger exchange area than totally clean heat exchanger case is required in order to reach similar performances. As a result, a good design would include outlet temperatures which exceed desired specifications when the exchanger is clean, unless bypass streams are provided. Fouling can occur as different mechanisms operating either alone or in combination, including corrosion, crystallization, decomposition, polymerization, sedimentation and even biological activity. Fouling factors are estimated from experiments and experience, even if the process to obtain them contains a lot of uncertainties and variables that are often not accounted for. The most comprehensive tabulation of fouling factors is the one developed by TEMA and it is in public domain.

The maintenance has different strategies depending on the type of heat exchanger. Plate and frame types can be periodically disassembled and directly cleaned, while for tubular ones it is necessary to use specific acids, sandblasting, high-pressure water jet, bullet cleaning and drill rods. Finally, some techniques help to prevent fouling, such as water treatments and addition of chemicals (Serth, 2007).

2.8 Optimization of a heat exchanger

All over the world, the process industry has to constantly face problems related to energy consumption, availability of resources as well as pollution discharge. These factors make the top priorities to be the efficiency improvement, the energy demand reduction and more severe controls of whatever emissions. Consequently, that means to reduce global costs and consumptions and to achieve green and efficient production, without forgetting the requested product quality specifications. In practice, for the discrete industry, represented by the mechanical manufacturing processes in general, this is a reachable target, but unfortunately for the process industry this is more difficult, as it is a continuous process of multiphase coexistence involving multiple physical and chemical reactions. Thus, it is hard to optimize and upgrade with intelligent, innovative technology.
In industrial processes heat exchangers are an essential part, since their competences are to produce, transfer and utilize the energy as well as to efficiently consume and recover it. So, their optimization plays a crucial role in improving the general performance of the chemical plant. They are often divided into groups, called heat exchange networks (HENs), in which each one is dependent on the others, like a big unit block. In the last 4 decades this subject was more and more studied as its technical improving, the research of higher efficiency and the need of saving energy became extremely important. The main goal is to find optimal systems from the energetic and costly point of view. Of course, a lot of heuristics and evolutionary steady-state synthesis methods are available in literature and allow to develop HENs, but often basing on fixed stream supply, flow rates and target temperature. However, dynamics plays a fundamental role, since there may be disturbances and uncontrolled upsets in upstream process units or simply the specifics needed by the market may change (Sun Lin, 2013).
In order to understand how to generally approach the optimization of a HEN, it's common to deal first with a single heat exchanger and then to extend the study to the definitive network.

2.8.1 Optimal design of a single heat exchanger

As already said above, the analyzed heat exchanger is classified according to its exchange methods and its structure, for example fixed tube, plate fin or floating head type. Each different kind shows determined transfer efficiency, compactness, weight and so on. In addition, the materials used have their own characteristics, like thermal conductivity, heat transfer resistance and heat transfer efficiency. Another important factor is the exchange area, which may affect a lot the final costs.
In general, the optimization is composed of parameters, design variables, objective functions and constraint conditions. In particular, the design variables are normally structural, such that flow rate partitions, fin or tube spacing, core length and width and number of channel layers. The constraints may be dimensional, that is the size of various pieces of the heat exchanger itself or the available

space, but they may be performance constraints too, which impose restrictions, for example, on the heat transfer efficiency and the pressure drop. Mathematically, the optimization is achieved by finding the maximum (or minimum) value of the objective function. The simple target optimization is the very simple type:

$$\min f(x)$$
$$g_i(x) \geq 0, \qquad j = 1,2,\dots,J$$
$$h_k(x) = 0, \qquad k = 1,2,\dots,K$$

However, it's more common to deal with a multi-objective combination optimization, which allows to have a wider view on the whole system:

$$\min \boldsymbol{f}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})\right)$$
$$g_i(x) \geq 0, \qquad j = 1,2,\dots,J$$
$$h_k(x) = 0, \qquad k = 1,2,\dots,K$$

where $f(x)$ is the objective function, $g_i(x) \geq 0$ are the inequality constraint and $h_i(x) = 0$ are the equality constraints (Yao, 2017).

2.8.2 Optimal design of a heat exchanger network

In industry, a lot of studies pays attention mostly on the optimization of the HENs, carrying out in-depth research and detailed analyses about that. In effect, some aspects are to be considered in order to have a satisfying overview of the entire system:

- The redistribution of the heat load of the existing transfer unit;
- The possibility to install one type of heat exchanger or another, depending on the requirements;
- The changes in the connections among all the heat exchanger units or in every design parameter that must match the previous costs;
- As some operative parameters are modified, the equipment must be changed as well, for example if the pressure drops overcome the allowable range, pumps and valves must be probably substituted or improved;

Thence, every HEN has its own combination of heat exchanger types, quantities, connections and flow distributions and the mathematical model needs to take into account all these parameters. Also, temperature, pressure, transfer coefficients, technical and environmental conditions must be considered. So, it's difficult to obtain a global mathematical model with a possible solution, but it

can be anyway established only if the parameters can be identified; and this happens for some HEN. The main variables to be considered are often the pressure drops, the heat transfer and both.

The control of the pressure drops is an important technical index, it can change after various adjustments to the whole HEN. Indeed, if the total exchange area becomes larger, the fluid resistance gets bigger, thereby increasing the pressure drops. When it is too large, the activity of pumps and compressors must be modified, that is their number or their power must increase, in order to ensure that fluid flow rates and other related indicators stay inside a reasonable range. Of course, this can result in additional equipment costs. The heat transfer is another crucial variable to be observed while modeling a HEN. In fact, when it must be increased, it's often necessary to adjust the global system, for example by adding heat transfer equipment or locate the existing one in different positions. That results in meeting the new requirements, but also in higher construction costs, complexity and even downtime. In this case, the enhanced heat transfer technology for the same conditions can reduce the area as well as the related expense and difficulties. When modifications are implemented, it's important to simultaneously control the pressure drops while strengthening the heat transfer.

Hence, the temperature of target streams of a HEN should be tightly controlled in order to manage both the safety and the economics of the whole plant. As demonstrated by many previous studies and by the experience too, placing bypasses throughout the HEN permits to satisfy the request of a good dynamic control, which is essential to achieve the aforesaid objectives (Yao, 2017).

2.9 By-pass method

As said above, heat exchangers are undoubtedly the most numerous industrial unit operation as the temperatures and the phases of process streams must be continuously changed to achieve a good degree of optimization. Hot sources of heat, like steam, hot oil and molten salts, and cold sinks of heat, like cooling or tap water, air and refrigerants, are utilized throughout different types of heat exchanger in order to heat, cool, vaporize or condense a process stream. By the way, process-to-process heat exchangers are widely used, because they allow to reduce that utility consumption, so that energy costs decrease. It's clear that most heat exchangers require some type of control system to achieve the desired process objectives. In particular, when a utility stream is involved, its flow rate is usually manipulated so that the process temperature stays under control; likewise, if there are phase changes, that is vaporization or condensation, the control occurs by keeping the vessel pressure or the liquid level fixed or at least inside a reasonable range.

Nowadays, many methods are used for controlling temperatures in heat exchanger systems and one of these is the bypassing, that is the direct manipulation of the flow rate of either the hot or the cold stream. This technique is especially used when that stream is a utility, therefore it helps the plant be less dependent on external fluxes. Another situation in which it is widely used is when the flow rates of both streams are set by requirements. Its principle is very simple: a portion of one of the streams, either hot or cold, is sent through the heat exchanger and the remainder is bypassed around the unit. Valves in each path control the mixed streams, providing very tight temperature control as the blending dynamics is very fast. The design optimization variables are essentially the fraction of bypassing, the area of heat transfer and the pressure drops over the valves.

There are many situations where bypassing brings a lot of advantages. For example, it is sometimes used in heat exchangers in which low velocities of cooling water could turn into fouling: the cooling water flow can be set at a fixed high flow rate and the hot stream is bypassed. Furthermore, when the heat exchanger involves a hot process stream and a cold one, their manipulation for control purposes is usually not possible, as their flow rates are set by upstream or downstream dynamics, so that bypassing is frequently employed. Other frequent applications of this bypass method deal with feed-effluent heat exchanges, condensers and pumparounds in distillation columns or heat-integrated systems.

There are basically two alternatives: bypassing the hot stream around the heat exchanger or the cold one. In general, the common heat-transfer heuristics is to bypass the stream whose outlet temperature is to be controlled. After the heat exchanger, the blending of both bypassed portion and the one through the unit occurs almost instantaneously, so that strict temperature control is definitely achievable. Actually, this fast mixing of both hot and cold stream quickly compensates for some secondary, slow disturbances caused by the flow rates passing through the heat exchanger.

One crucial characteristics of bypass method is the trade-offs between the partitioned streams. In fact, it is observable that increasing the bypassed portion of flux permits larger variations of heat transfer dynamics, so that it is easier to reach changing process requirements, but unfortunately it also requires a larger exchange area, as the differential temperature driving forces become smaller. There are other situations like this in which a sort of trade-off is really visible, hence this is a clear demonstration of the interaction between design and control, that can even turn into a conflict.
Another important issue to be counted is the design pressure drop through the heat exchanger, which is evaluated by heuristics with the purpose of giving reasonable heat transfer parameters. Indeed, if it increases, it generates higher velocity of the fluxes, larger film coefficients and smaller exchange area, thus obtaining lower capital investments. However, at the same time it means that pumps and compressors must work more intensively and the energy requirements of motors and turbines increase as well, giving higher operative expense. The importance of this issue is also linked to the control valves, which are usually applied on the bypass stream only and not on the other one (as shown in figure 4A). As a matter of fact, the pressure drops should be the same in both the bypass line and the heat exchanger one, but in practice the rangeability is often very limited, especially when the valve is opened wide because there is still flow through the unit. Therefore, it is more logical to install two control valves, one on the bypass and one on the other stream (like in figure 4A), so that it is more manageable to adjust those two flow rates and the possible ratio of maximum-to-minimum heat transfer is wider. Anyway, in most of situations, the functioning of the heat transfer system depends on what happens upstream or on what are the global requirements of the entire plant, so that it could be harder to match the optimal conditions. Many times, to face these problems, an upstream variable-speed pump is used in some liquid systems to maintain the total required flow rate; otherwise, one valve may control the total flow rate while the other one may deal with temperature (see figure 4B). It's important to notice that, in general, every valve amplifies pressure drops and motor work demands (Luyben, 2011).

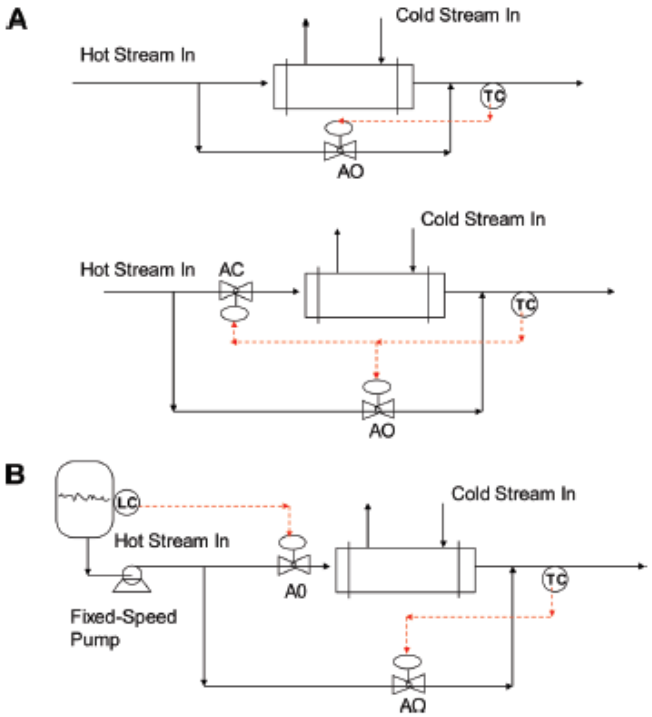


Figure 12. (A) Heat-exchanger bypass with and without valve in heat exchanger line. (B) Heat-exchanger bypass with fixed-speed pump. (Luyben, 2011)

2.10 ALAMO (Automated Learning of Algebraic Models for Optimization)

ALAMO (Alison Cozad, Automated Learning of Algebraic Models for Optimization, 2014) is a computational methodology for learning algebraic functions from data. Basically, a low-complexity, linear model is built on explicit non-linear transformations of the independent variables, whose linear combinations allow to better approximate complex behaviors observed in real processes. There are two fundamental properties that make this software very efficient: the first is its model refining through an error maximization sampling, which can be called adaptive sampling; the latter is the derivative-free optimization. Moreover, constraints on the response variables can be enforced in ALAMO, increasing its accuracy.

ALAMO works with a regression and classification model learning methodology through which it builds simple, accurate surrogate models starting from a minimal set of sample points, coming from experiments, simulations or other sources. The main instrument exploited at the heart of this software is an integer-programming-based technique that considers, at the same time, more explicit transformations of the original input variables x. Afterwards, this crude model is tested and improved by derivative-free optimization solvers by means of an adaptive sampling.

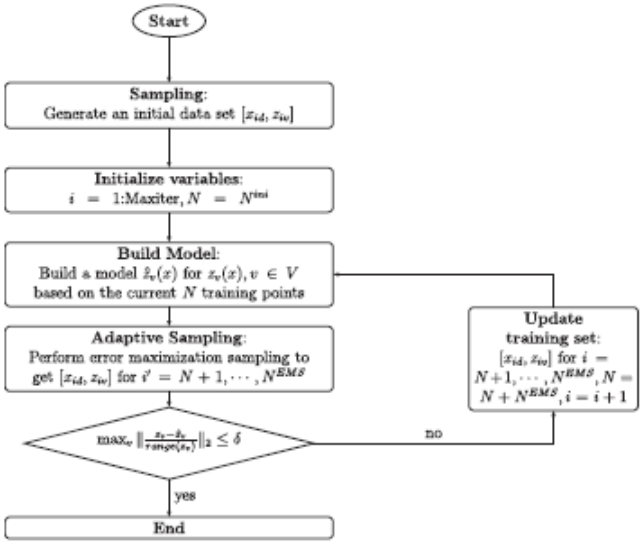


Figure 13*. Flowchart of the model building algorithm applied by ALAMO. (Sahinidis, 2017)*

More in particular, ALAMO is considered a linear parametric regression, where explicit parametrization of the original input into the system are exploited to accurately describe any non-linear behavior. Generally, the software uses a list of default transformations, such as monomial, binomial, ratio, exponential, logarithmic and trigonometric. Additionally, more complex transformations can be selected, such as sigmoid, Arrhenius and Gaussian relationships, consequently bringing to a hyper parametrization or even to a non-parametric regression field.

In order to explain how ALAMO acts in the phase of learning surrogates, a Mixed Integer Problem (MIP) formulation is used to optimize a model fitness metric by exploiting a parametrical minimization of the sum of square residuals (SSR) to a maximum number of non-zero regression coefficients, even though non-linearities are often present. In particular, the general form of the SSR is:

$$SSR = \sum_{i=1}^{N} \left(z_i - \sum_{j=1}^{k} \beta_j x_{ij} \right)$$

However, our goal may be obtained by defining a series of Cardinality Constrained Mixed-Integer Quadratic Programs (CCMIQP), whose formulation is:

$$\min_{r=1,\dots,k} FM(r)$$

where

$$FM(r) = \min \sum_{i=1}^{N} (z_i - X_i\beta)^2 + C(r)$$

$$s.t. \sum_{j=1}^{k} y_j \leq r$$

$$-My_j \leq \beta_j \leq My_j, \qquad j = 1, \dots, k$$

$$y_j \in \{0, 1\}, \qquad j = 1, \dots, k$$

As seen in this general case, a set of N training points is taken, where each data point contains a set of input data x_{id} (with $d \epsilon D$) related to a set of responses z_{iv} (with $v \epsilon V$). Considering a single response vector for simplicity, an amount of non-linear transformations is performed in order to fill in a regression design matrix X_{ij} , $j = 1, \dots, k$. In particular, the use of a MIP formulation allows to balance the bias-variance trade-off, which is typical of every technique working on whatever data set. FM is any model fitness metrics, available in various different forms in literature, as shown in one of the previous tables. $C(r)$ is a metric-dependent complexity penalty, that is, a constant for a well determined model cardinality r. In fact, defining a cardinality constraint on the binary variables y_i together with big-M constraints it's sure that no more than r regressors are included in the model. The constants M are similar to the ones used in the Lasso regression:

$$M = \sum_{j=1}^{p} |\hat{\beta}_{j,ols}|$$

However, for the Cp, BIC and RIC model fitness metrics a convex MIQP formulation for a direct optimization can be exploited:

$\min FM$
$s.t. -My_j \leq \beta_j \leq My_j \, , j = 1, \dots, k$
$y_j \in \{0,1\} \, , j = 1, \dots, k$

These two problem formulations are analogous, although the previous one takes into account just k/r models under the cardinality constraints, while the latter one refers to all 2^k possible regression models. Moreover, they are similar, but not totally equivalent to the penalized version of the lasso problem as well as to constrained one, whose characteristics are different. In general, the constrained lasso problem is written as:

$$\left\{ \min_{\beta} SSR \,\middle|\, \|\beta\|_1 \leq T \right\}$$

where the condition on the parameter $T < \left\|\hat{\beta}_{ols}\right\|_1$ causes a shrinkage in the regression coefficients. Instead, a penalized lasso problem can be written as:

$$\left\{ \min_{\beta} SSR + C\|\beta\|_1 \right\}$$

where C is a trade-off parameter, that has a maximum value where all regression coefficients are zero. This null solution is often the starting point from which smaller C parameters are found through the typical warm-starting process. This penalized version is more used than the other one, due to the more efficient computational implementations found till now. Therefore, between CCMIQP and MIQP formulation the first has a direct control on the number of non-zero regression coefficients, but it results in computational inefficiency when facing big amounts of regressors, while the latter shows solutions more rapidly.

2.10.1 Adaptive sampling

An adaptive sampling technique (or active learning) is a procedure for selecting units which depends on the previous observations made during the study. Basically, the underlying response surface is obtained by exploring the object system at desired input levels. Compared to the traditional fixed sampling procedure, adaptive sampling techniques often lead to more effective results. In fact, it's

impossible to identify a priori a data set which perfectly train a model (even called design of experiments DOE), but some methodologies can somehow fill the space of the input variables in an optimal fashion. Some examples of this technique include factorial designs, Latin hypercube and orthogonal arrays, such as systematic sampling. From a general point of view, they simply work in three steps:

1. Generation of a set of data points in the domain of the input variables;
2. Evaluation of these points;
3. Moving on to train the model.

Therefore, these methods are fast, not difficult to implement and computationally efficient, nevertheless the produced data set may not be optimal to train a design of experiment. In effect, specific problem areas in the domain may be underrepresented causing the global fidelity of the resultant model to decrease. Anyway, these are iterative techniques, where current model and data set are exploited to find new promising points. They are divided into two types:

1. Exploration-based iterative methods, working in a single-pass way, that is, they fill in the space of the input domain at each iteration.
2. Exploitation-based iterative methods, which locate data points near regions that are difficult to model, such sharp non-linearities or discontinuities.

The peculiar characteristic of every active learning is the error maximization sampling (EMS) approach:

$$\max_{x^l \le x \le x^u} \left(\frac{z(x) - \hat{z}(x)}{z(x)} \right)$$

where the output data $\hat{z}(x)$ is obtained by ALAMO, while true value $z(x)$ is formally unknown. Then, it's mandatory to use any DFO solver, such as SNOBFIT, whose effectiveness is demonstrated in (Huyer W., 2008). As a rule, it works with few steps:

1. It estimates the error at new candidate points;
2. It compares it with a region of model mismatch, so that if the new point is found to be inside there, then it's added to the training set and consequently it generates a new adaptive model;
3. These added points are used to evaluate the true value of the true value $z(x)$, in order that the EMS can be found;
4. If the EMS is larger than a tolerance δ, then the model must be retrained again with the updated data set; instead, if the EMS is smaller than δ, then there is the convergence on the final surrogate model, since $z(x)$ and $\hat{z}(x)$ are practically equal.

2.10.2 Constrained regression

Most of times, upper and lower bounds on the regression variables are used, expressed with the general form:

$$\min_{\beta \epsilon A \cap \Omega(\chi)} g(\beta; x, z)$$

where $\beta \epsilon \mathbb{R}^k$ are the regression parameters and a given loss function g is minimized over an original set of regression constraints A. Moreover, $\Omega(\chi)$ is an additional set of constraints able to enforce the minimization:

$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^k : f[x, \hat{z}(x; \beta)] \leq 0, \qquad x \in \chi\}$$

in which the function f is a constraint in the space $\mathbb{R}^k$ of the modeled responses $\hat{z}(x)$ and the transformed predictors x, while χ is a nonempty subset of $\mathbb{R}^n$. It's clear that the feasible domain where the model prediction is applied is just the intersection between A and $\Omega(\chi)$ and this is very useful to reduce the working area for any regression analysis.

In the literature, it's been demonstrated the existence of relationships between the regression coefficients and they result in the form of inequalities. They are often important because they can simplify the model prediction when they are known a priori. Nevertheless, for this work it's more convenient and feasible to obtain a constraint based on the estimated response of the regression model and guaranteed across a chosen domain. The problem is that $\Omega(\chi)$ must be valid over the full domain of the inputs, meaning that the inequalities expressions are to be applied to infinitely many points, that is $\forall x \epsilon \chi$. The main risk is to look for countably many regression variables, but an infinite number of constraints for every single equation f, therefore an impossible problem to be solved. That's why semi-infinite programming (SIP) techniques must be implemented, that is, optimization models which work with a finite number of variables, but an infinite number of constraints. Basically, their target is to solve $\max_{x \epsilon \chi} f[x, \hat{z}(x; \beta)]$ to find points in the domain where both maximum violation in the current surrogate model and the applied constraint occur, considering that $\beta \epsilon \Omega(x)$ if and only if $\max_{x \epsilon \chi} f[x, \hat{z}(x; \beta)] \leq 0$. In general, SIP techniques are non-linear and non-convex.

The constraints allow the user to:

1. Restrict the bounds of the response, that is, lower and upper ones, individually or altogether, then reducing the computational efforts;
2. Restrict the derivatives of the response enforcing the model convexity or a monotonic relationship, increasing the accuracy;
3. Enforce the bounds over a domain, even when it's outside of the training domain, improving the flexibility. It's important to observe that this is an alternative to the classical blind extrapolation outside the given data set region, because the bounds on the response are always certified for the selected model, therefore it acts a safer way.

In order to explain how SIP methods work, it's possible to analyze the two-phase general one, easier to understand. The algorithm showing how it solves the semi-infinite constrained regression problem is represented here below:

Given a training set of dependent and independent data points $[x_{id}, z_{ik}]$ for $i = 1, \ldots, N$, $d = 1, \ldots, n$, $k = 1, \ldots, m$; requested feasible points n_{viol}; and relevant tolerance values

Initialize $l = 0$ and χ^0 as ø or by using a design of experiments
while $(f(x^l) > \epsilon)$ or $(l < 1)$ **do**
  $\beta^l \leftarrow$ solve (PI)
  $x^l, f(x^l) \leftarrow$ find n_{viol} isolated feasible points for (PII_{feas})
  if $x^l = = \emptyset$ **then**
    β^l is both feasible and optimal to (C)
    break
  else
    Update feasible set, $\chi^{l+1} \leftarrow \chi^l \cup x^l$
  end if
  $l \leftarrow l + 1$
end while

Figure 14*. Algorithm to solve semi-infinite constrained regression problem. (Alison Cozad, 2014)*

In the Phase I (PI) a relaxation of the object problem is solved:

$$\min_{\beta \epsilon \mathcal{A} \cap \Omega(\chi^I)} g(\beta)$$

where the parametric constraint $g(\beta)$ acts over the finite subset $x \epsilon \chi^I \subset \chi$. In this step an approximation of the regression coefficients β^I is found over the relaxed feasible region, defined now as:

$$\Omega(\chi^I) := \{\beta \epsilon \mathbb{R}^m : f[x, \hat{z}(x;\beta)] \leq 0, \qquad x \in \chi^I\}$$

Then there is the Phase II (PII), where the maximum violation problem is solved:

$$\max_{x \in \chi} f(x, \hat{z}(x; \beta^I))$$

The solution points β^I are used to solve (PII). The algorithm stops as soon as the solution x^I from P(II) satisfies the constraint $f(x^I) \leq 0$, otherwise it keeps going on by updating each time the new set of data points $\chi^I = \chi^I \cup x^I$ to be implemented in Phase I. It's observable that (PI) tends to preserve the convexity of the original regression problem [original regression problem (U)], except for some cases. So, when the problem is a linear regression, then the feasible region $\Omega(\chi^I)$ is linear too; nevertheless, the (PII) is very often non-convex, so that a global optimization solver is needed. Of course, by increasing the problem size the complexity of the regression becomes more important and more resource intensive: this happens augmenting the number of basis function for the (PI) or considering more predictor variables x for the (PII).

To solve (PII) a multi-start local search would lead to a lot of iteration, therefore a branch-and-reduce optimization is applied to guarantee rigorous global optimality. In addition, several isolated feasible solutions for PII are evaluated in order to try to decrease the total constrained regression iterations.

The starting point of the aforesaid algorithm is a set of feasible points $\chi_0 = \emptyset$, so it can be empty, that is it can be equal to 0, or nonempty, for example selected by previous samplings. By solving (PI), an initial approximation β^0 is found, then it's used to calculate ($\beta^I = \beta^0$), which allows to locate up to n_{viol} isolated feasible points of the problem PIIfeas:

$$\max_{x \in \chi} f(x, \hat{z}\,(x; \beta^I))$$
$$s.t.\ f\big(x, \hat{z}(x;\ ; \beta^I)\big) - \epsilon_{viol} \geq 0$$

It's important that the chosen feasible points don't overlap each other, so that they cannot be redundant: that's why they are selected such that $\left\|x_i^0 - x_{i'}^0\right\|_\infty \geq \epsilon_{isol}$ for every possible pair of points i and i'. Moreover, there should be two characteristics: first, the objective function should reflects the magnitude of violation, allowing the solver to locate the set of isolated feasible points with large, ranked violations; second, the points with $f\big(x, \hat{z}(x;\ ; \beta^I)\big) = 0$ should not be considered, even if they are compatible with the original minimization problem (C), so that a strict violation can be obtained with a smaller ϵ_{viol}. At the end of each step $\chi^{I+1} = \chi^I \cup x^I$ is updated and (PI) is solved again on a new region $\Omega(\chi^{I+1})$. The iterations proceed until the current regression parameters β^I

are both optimal and feasible for (C). Of course, β^I must be the solution of PI and the maximization $\max_{x\in\chi} f\left(x, \hat{z}(x;\ ;\beta^I)\right) \leq 0$.

2.10.3 Classes of constrained regression in the x- and z- domains

The structure of these problems receives benefits just by constraining the original space of the system.

Restricting individual responses

Each constraint is individually applied on the response variables, thus obtaining a semi-infinite feasible region:

$$\Omega(\chi) := \{\beta\epsilon\mathbb{R}^m: a\hat{z}(x;\beta) + h(x) \leq 0, \qquad x \in \chi\}$$

where $h(x)$ is a function of predictors x and $a \in \mathbb{R}$ is a coefficient which gives a sort of scale of the response $\hat{z}(x)$, while $h(x)$ is the order of the constraints.

Zero-order restrictions, that is, $h(x) = 0$, can be used when upper or lower bounds on $\hat{z}$ are logically present, for example flow rates, geometric dimensions or absolute pressure, which cannot have negative value. Indeed, the most common and intuitive lower bound is the non-negativity of the response variables. Other objective functions must be searched inside well-defined limits, like compositions or probability distributions, that must be in the range from 0 to 1. The formulation of this problem can be expressed in Phase I and Phase II as it follows:

$\left(PI^{bnd}\right)\ \min_{\beta\in\mathcal{A}} g(\beta)$

$s.t.\ \ \hat{z}(x;\beta) \leq z^{up} \quad x \in \chi^I$

$\hat{z}(x;\beta) \geq z^{lo} \quad x \in \chi^I$

$\left(PII_{feas}^{bnd}\right)\ \min_x \hat{z}(x;\beta^I)$

$s.t.\ \ \hat{z}(x;\beta^I) \leq z^{lo} - \epsilon_{viol}$

$x \in [x^{lo}, x^{up}]$

$\left(PII_{feas}^{upbnd}\right)\ \max_x \hat{z}(x;\beta^I)$

$s.t.\ \ \hat{z}(x;\beta^I) \geq z^{up} + \epsilon_{viol}$

$x \in [x^{lo}, x^{up}]$

Z_{up} is the upper bound and z_{lo} is the lower one, while the problem space of the original predictors for any chosen upper and lower limit on x is defined as $\chi = [x_{lo}, x_{up}]$.

Instead, when $h(x) \lesseqgtr 0$, it's nonzero-order restrictions problem, hence there are non-constant constraints on a response variable $\hat{z}$. Of course, it's mandatory that the modeler knows more information in order to be reliable, but the model can consequently gain accuracy and robustness. Those constraints can be linear or not and, in practice, they can be initial and boundary conditions, mass and energy balances or intrinsically limited problem. In engineering, these levels of knowledge are often available for a simplified system, which can therefore be used as a further bound for the model. In fact, if this simple system represents the theoretical limit, not just an approximation, then it can be exploited to bound the output. For example, for heat transfer it's possible to bound the global system using the information derived from a previous study on the Carnot engine. Again, in reactor design, if it's available the knowledge of a simple model for a reactive system with no by-products, it can be the lower bound for the concentration of the whole process.

Restricting multiple responses

It occurs when simultaneous additional relationships between response variables $z_k, k = 1,2, \dots, n_{resp}$ are available, so that the feasible region can be represented as:

$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^m : d(\hat{z}(x;\beta)) + h(x) \leq 0, \qquad x \in \chi\}$$

where d is a function for the restrictions among all responses $k = 1, 2, \dots, n_{resp}$. For example, these relationships may be the mass or energy balances on inlet or outlet flows with two or more flows being response models; otherwise, they may be discretized state variables, such as fluid velocity profile along a tube, resulting in an intrinsic order of modeled variable values in the form $\hat{z}_{k'} \leq \hat{z}_{k>k'}$. Apparently, the solution of this type of problem is not compatible with ALAMO, because all its response models should be simultaneously solved. Since ALAMO needs to treat all the model output variables independently, an adaptation must be applied. A general case is considered:

$$\left(PI^{mult}\right) \quad \min_{\beta \in \mathcal{A} \cap \Omega(\chi^I)} g^{mult}(\beta_k)$$

where g^{mult} represents the objective of the simultaneous regression problem.

It's necessary to exploit a weighted linear least squares regression with weighting factors w_k:

$$g_{mult}(\beta_k) = \sum_{k=1}^{n_{resp}} w_k \sum_{i=1}^{N} \left(z_{ki} - \sum_{j \in \mathcal{B}} \beta_{kj} X_{ij} \right)^2$$

The algorithm that solves this general problem is shown here:

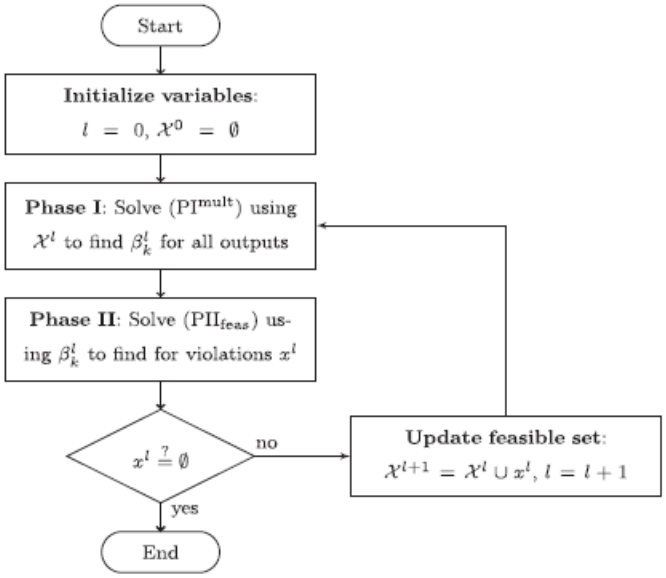


Fig. 3. Restricting multiple responses for the general case.

Figure 15*. Restricting multiple responses for the general case. (Alison Cozad, 2014)*

For the application of this algorithm to ALAMO, instead of fixed functional forms for response variables, more flexible ones are used after the implementation of $\Omega(\chi)$. Thus, the software overcomes the obstacle of concurrent multiple restrictions by managing to solve for each response $z_{k'}$ independently, using previous iteration's surrogate models for $z_{k \neq k'}$. So, the problems Phase I and Phase II become as follows:

$$\left(PI_{k'}^{mult}\right) \quad \min_{\beta_{k'} \in \mathcal{A}_{k'}} g(\beta_{k'})$$

$$s.t. \quad d\left(\hat{z}_{k'}(x;\beta_{k'}), \hat{z}_{k \neq k'}\left(x;\beta_k^{l-1}\right)\right) + h(x) \leq 0, \qquad x \in \chi^l$$

$$\left(PII_{feas}^{mult}\right) \quad \max_x d\left(\hat{z}_k\left(x;\beta_k^l\right)\right) + h(x)$$

$$s.t. \quad x \in [x^{lo}, x^{up}]$$

where g_k and $\mathcal{A}_k$ are given (A) for response variables $k = 1, 2, \dots, n_{resp}$. $PI_{k'}^{mult}$ must be solved for each k and it can happen that after the regression of each individual response variable $\beta_k^I \notin \Omega(\chi^I)$, since $PI_{k'}^{mult}$ is solved using previous models for responses $k \neq k'$. That's why the functional form of each response is fixed to ensure the feasibility of the resulting model after each iteration and PI_{feas}^{mult} is solved using a linear least squares regression, as illustrated above for $g_{mult}(\beta k)$. By doing so, $\beta_k^I \notin \Omega(\chi^I)$ is assured each time at the end of Phase I and before managing the Phase II. The entire algorithm adapted and applied to ALAMO is shown here below:

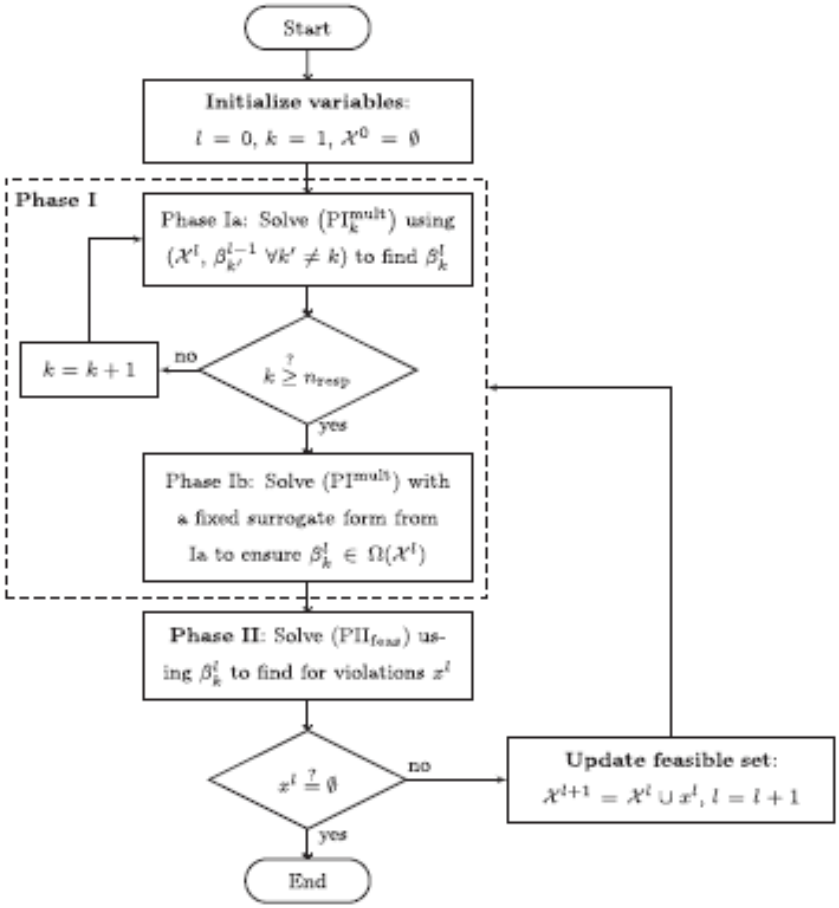


Figure 16*. Extending restricting multiple responses to ALAMO. (Alison Cozad, 2014)*

Restricting responses derivatives

When the knowledge about the system is enough, restrictions related to pre-existing derivative or partial derivative information combined to first-principles theory data may be very advantageous. Depending on the type of enforced constraints used, the regression model must be once- or twice-differentiable, so that the restricted feasible regions on first and second derivatives of each response model $\hat{z}$ with respect to the predictors x have a functional form like the following:

$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^m : a^T \nabla_x \hat{z} + h(x) \leq 0, \qquad x \in \chi\}$$
$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^m : a^T \nabla_x^2 \hat{z} + h(x) \leq 0, \qquad x \in \chi\}$$

Normally, derivative constraints are the most elegant and complete combination of empirical data, first principles, experience and intuition altogether, hence one of the best ways to face a mathematical problem in general since there many initial instruments that may be used for its solution. A common type of first derivative restrictions is the monotonicity of a response variable with respect to one or more predictors: for example, cumulative distributions, entropy of enclosed systems, gas pressure depending on temperature and under ideal or non-ideal conditions or imposing initial and boundary conditions.

However, when second derivative restrictions are implemented, the modeling is surely more complicated, but it may be extremely useful for regressions, since mathematically they are strictly connected to the convexity and the concavity. In effect, if data comes from a convex distribution, automatically the resulting should be convex itself. It's a sort of enforcement of the derivative bounds, that can help a lot in an optimization contest and it's obvious that the modeler must know a priori that information. In addition, the regression model $\hat{z}(x)$ must be twice differentiable and the Hessian matrix of partial second derivatives $H(\beta, x)$ must be positive-definite. This means that its determinant is compulsory non-negative:

$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^m : \det(H(\beta, x)) \geq 0, \qquad x \in \chi\}$$

Through an example, it is possible to better understand how to ensure the convexity of a surrogate model $\hat{z}$ previously obtained. We consider that the expression of $\hat{z}$ we ended up with is:

$$\hat{z}(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2^2 + \beta_4 x_1^2 x_2$$

Hence, its relative Hessian matrix is:

$$H(\beta, x_1, x_2) = \begin{pmatrix} 2\beta_4 x_2 & 2\beta_3 x_2 + 2\beta_4 x_1 \\ 2\beta_3 x_2 + 2\beta_4 x_1 & 2\beta_3 x_1 \end{pmatrix}$$

Then, a restriction of the determinant of H to the nonnegative space is applied in order to enforce the desired convexity on the feasible region of the regression and it is nonlinear and nonconvex in the β space. With this operation, a quadratic problem (QP) is automatically transformed into a quadratically constrained quadratic problem (QCQP), by means of least squares regression objective, so that the result is:

$$\Omega(\chi) := \{\beta \epsilon \mathbb{R}^5 : -\beta_3 \beta_4 x_1 x_2 - \beta_4^2 x_1^2 - \beta_3^2 x_2^2 \geq 0, \qquad x \in \chi\}$$

2.10.4 Safe extrapolation

Until now, carious types of restrictions were applied on the response variables over the range of the original predictor variable, in order to obtain an improving efficiency of surrogate modeling. Yet, it is possible to obtain other good results by including an expected extrapolation range, that is by expanding the enforcement domain. Normally, it's not a common technique, because it's quite dangerous and not very reliable, but in engineering it is somehow used to forecast results beyond an initial sample space. Indeed, if extrapolation is used while paying attention, it increases the likelihood of prediction error. So, the set of input points $x \in \chi_{extrap}$ may be expanded and at the same time any of the aforesaid problem classes may be implemented, improving the accuracy of both extrapolation and original problem domain.

2.10.5 Boundary and initial conditions

While modeling empirical data, boundary conditions are imposed on ordinary or partial differential equations (ODE or PDE), which are subdivided into three types: Dirichlet, Neumann and Robin (for more details, see (Sandro Salsa, 2013))

As it can be seen, Dirichlet boundary conditions specify the value of the solution z at a fixed location x_i^* of the x-domain and the feasible region $\Omega(\chi^I)$ they are applied on is similar to that of restricting individual response. Instead, Neumann ones specify the value of the gradient at a fixed location in at least one dimension x^{i^*}, while Robin ones specify a linear combination of function values and derivatives at a fixed location in the domain. For these two types the feasible region $\Omega(\lambda^I)$ of application is more similar to that of restricting multiple response. Anyway, all these boundary

conditions are enforced in a reduced dimensional space and the result is often a more physically consistent regression model. A common example is when initial conditions are about the time domain $t = 0$ without restricting the space domain: its dimensions x_i are reduced by one or more to $\chi \cap \{x_i = x_i^*\}$. In other words, this domain is contracted by enforcing conditions over a subset of χ. It's important to consider that standard boundary conditions require parametric equality constraints, obtaining a reduction of the feasible region, but also an obstacle to convergence of the optimization. That's why it's provided a tolerance ϵ on the slack of equality for $a, b \in \mathbb{R}^n$, exactly like it's shown here below:

$$\Omega(\chi) := \left\{ \beta \in \mathbb{R}^m : \begin{matrix} \hat{z}(x;\beta) + a^T \nabla_x \hat{z} + b^T \nabla_x^2 \hat{z} - h(x) \leq \epsilon_{viol} \\ \hat{z}(x;\beta) + a^T \nabla_x \hat{z} + b^T \nabla_x^2 \hat{z} - h(x) \geq -\epsilon_{viol} \end{matrix} \; x \in \chi \cap \{x_i = x_i^*\} \right\}$$

These restrictions allow the regression model to gain higher fidelity and to use simulation data and first-principles boundary conditions.

3 METHODOLOGY

3.1 Design of a heat exchanger

3.1.1 Approach to heat exchanger design

The design of a heat exchanger includes all the knowledges about heat transfer and they must be used in a proper way: that's way it could be considered an art. Of course, it is mandatory to strictly keep in mind the unavoidable differences between ideal conditions and real dynamics of every system and its environment, thus involving its design too. In effect, the result of this modeling must satisfy process, operational and economical requirements, such as availability, flexibility, maintainability and sustainability. However, there is an intrinsic error in any design, in particular in any passage from one design level to the successive one, for example passing from the basic scheme to equipment, to the process: for this reason, heat exchanger design is not a highly accurate art under the best conditions. Anyway, the general guideline is composed of the following steps (Perry, 1999):

1. Specification of process conditions, such as stream compositions, flow rates, temperatures and pressures, that are practically the boundary conditions;
2. Evaluation of the required physical properties over the temperature and pressure ranges;
3. Choice of the type of heat exchanger;
4. Preliminary estimation of the size of the heat exchanger by implementing an appropriate heat transfer coefficient for fluids, process and equipment;
5. Choice of a first-level design, which contains all the details necessary to carry out the successive design computations;
6. Evaluation, or rating, of the design in the previous step and verification of its ability to undergo process specifications related to both heat transfer phenomenon and pressure drop;
7. Choice of a new configuration of heat exchanger if the design at step 6 does not meet the requirements or it is inadequate, so step 6 is basically repeated iteratively until the target is reached. It is usually necessary to enlarge heat exchanger, or even consider multiple-exchanger configurations, while maintaining specifications and similar feasible limits of pressure drops, tube length, shell diameter, etc. On the contrary, if the design at step 6 is over-dimensioned, that is the heat load is bigger than the requested one or it does not exploit all the allowable pressure drop, then it is possible to pass to a less expensive design;
8. At this final step, the design should meet process requirements at lowest possible cost and with reasonable expectations of error. Of course, the cost should include capital and installation expense, operation and maintenance costs as well as credits to face long-term process. modifications. Normally, the best selection for heat exchangers is not automatically based on a lowest-price condition, as this strategy frequently results in future penalties.

3.1.2 Overall heat transfer coefficient

The very first equation on which heat exchanger design is based is:

$$dA = \frac{dQ}{U\Delta T}$$

whose meaning is quite simple: dA is the element of surface through which an amount of heat dQ is transferred at a point in exchanger where U is the overall heat transfer coefficient and ΔT is the overall bulk temperature difference between the two exchanging streams. U is related to individual film heat transfer coefficients h_i, fouling factors R_i and wall resistances, including heat conductivities k_i; hence, basing $U_o$on the outside surface area A_o, the most general expression of overall heat transfer coefficient is the following:

$$U_o = \frac{1}{1/h_o + R_{do} + xA_o/k_w A_{wm} + (1/h_i + R_{di})/A_i}$$

In order to obtain the total outside area required to transfer the total heat load Q_T, the first expression can be integrated as shown below:

$$A_o = \int_0^{Q_T} \frac{dQ}{U_o \Delta T}$$

The above integration can be computed only if $U_o$and ΔT form is known. In effect, they are both functions of Q; in addition, U_o always varies strongly and nonlinearly throughout the exchanger. Thus, U_o and ΔT must be necessarily evaluated at several values and numerically or graphically integrated. In practice, a constant mean value U_{om} can be estimated by means of the above expression and a mean ΔT can be defined, so that a reliable approximation of A_o can be found (Perry, 1999):

$$A_o = \frac{Q_T}{U_{om}\Delta T_m}$$

Some attentions should be taken:

- U_o does not vary too strongly along heat exchanger coordinate;
- The equations and conditions chosen to estimate individual coefficients are the proper ones;
- The mean temperature difference for the designed exchanger configuration is correct.

3.1.3 Mean temperature difference

It is clearly observable that temperature difference between the two fluids will generally vary at every consecutive point along the heat exchanger. Therefore, the mean temperature difference ΔT_m (also called MTD) may be evaluated from the terminal temperatures of both streams, but considering the following assumptions as valid:

1. All elements of a given fluid stream have the same thermal history through the heat exchanger;
2. The system is in steady-state condition;
3. Specific heat is constant for each stream or if there is a phase transition in both streams;
4. The overall heat transfer coefficient is constant;
5. Heat losses are negligible.

Countercurrent or cocurrent flow

There are some common cases in which the correct temperature difference to apply is not exactly the aforesaid MTD, but the Logarithmic Mean Temperature Difference, also called LMTD, as it describes more accurately the gap of temperature between the two streams at every point. Those cases occur when flows completely countercurrent and cocurrent or when one or both streams are isothermal, for example condensation or vaporization of a pure component with negligible pressure change. Here below the definition of LMTD for countercurrent flow:

$$LMTD = \Delta T_{lm} = \frac{(t_1' - t_2'') - (t_2' - t_1'')}{\ln(\frac{(t_1' - t_2'')}{(t_2' - t_1'')})}$$

Instead, here is the LMTD for cocurrent flow:

$$LMTD = \Delta T_{lm} = \frac{(t_1' - t_1'') - (t_2' - t_2'')}{\ln(\frac{(t_1' - t_1'')}{(t_2' - t_2'')})}$$

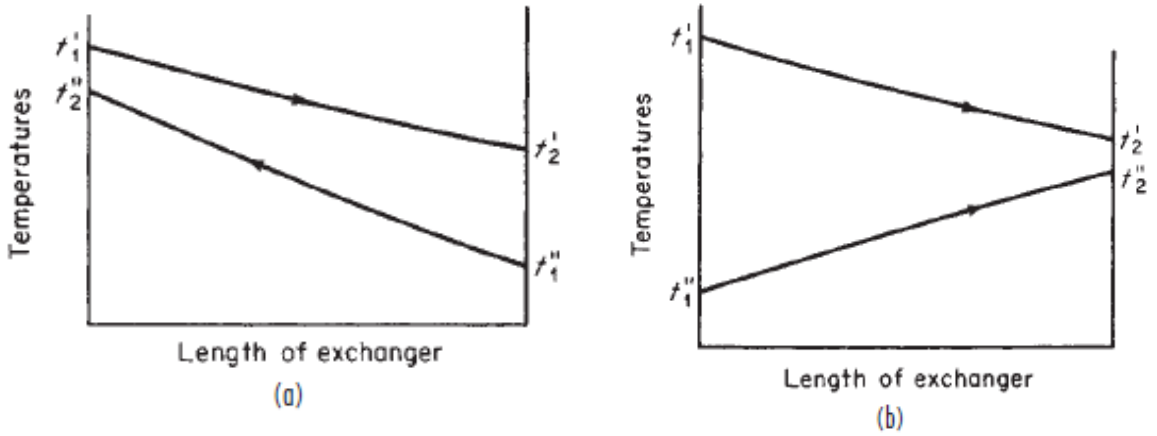


Figure 17. Temperature profiles in heat exchangers. (a) Countercurrent. (b) Cocurrent. (Perry, 1999)

There is a connection between temperature difference ΔT and overall heat transfer coefficient U: indeed, if U is not constant but linearly dependent from ΔT, it is possible to directly evaluate the quantity $U_{om}\Delta T_m$ for countercurrent flows by means of the following expression, presented in literature by Colburn:

$$U_{om}\Delta T_m = \frac{U_0''(t_1' - t_2'') - U_0'(t_2' - t_1'')}{\ln\left(\frac{U_0''(t_1' - t_2'')}{U_0'(t_2' - t_1'')}\right)}$$

where U_0'' is the overall heat transfer coefficient evaluated when the stream temperatures are t_1' and t_2'' and U_0' is evaluated at t_2' and t_1''. An analogue quantity $U_{om}\Delta T_m$ relative to cocurrent flows is given by a very similar expression:

$$U_{om}\Delta T_m = \frac{U_0''(t_1' - t_1'') - U_0'(t_2' - t_2'')}{\ln\left(\frac{U_0''(t_1' - t_1'')}{U_0'(t_2' - t_2'')}\right)}$$

where U_0' is evaluated at t2' and t_2'' and U_0'' is evaluated at t1' and t1''. Of course, the two value U_o must be estimated in advance in order to implement these equations above; in effect, when a hydrocarbon stream is the limiting resistance, some caloric temperature charts developed by Colburn are available in literature. Anyway, those equations usually give satisfying results even though the approximation of U_o being linear with ΔT is not totally exact (Perry, 1999).

3.1.4 Reversed, mixed or cross-flow

Sometimes the flow pattern cannot be completely defined as countercurrent or cocurrent, so the LMTD may be multiplied by a correction factor *FT* in order to attain the proper MTD. These parameters have been mathematically derived during previous studies for the most common flow patterns of interest and they are largely available in literature as particular charts to be consulted; In Figure 19, there are some examples taken from "Perry's Chemical Engineers' Handbook":

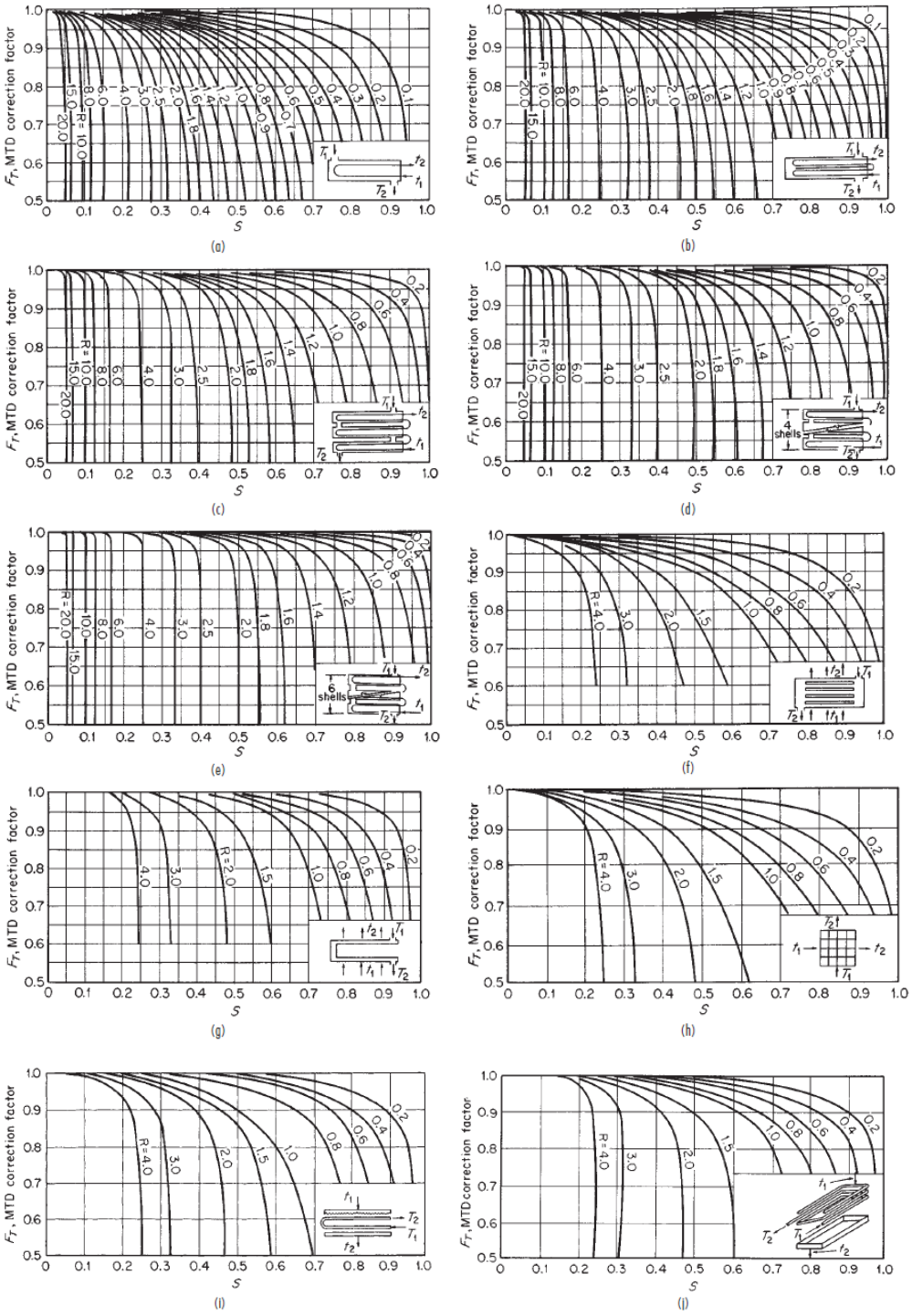


Figure 18. LMTD correction factors for most commonly used heat exchangers. (*a*) One shell pass, two or more tube passes. (*b*) Two shell passes, four or more tube passes. (*c*) Three shell passes, six or more tube passes. (*d*) Four shell passes, eight or more tube passes. (*e*) Six shell passes, twelve or more tube passes. (*f*) Cross-flow, one shell pass, one or more parallel rows of tubes. (*g*) Cross-flow, two passes, two rows of tubes; for more than two passes, use $FT = 1.0$. (*h*) Cross-flow, one shell pass, one tube pass, both fluids unmixed (*i*) Cross-flow (drip type), two horizontal passes with U-bend connections (trombone type). (*j*) Cross-flow (drip type), helical coils with two turns. (Perry, 1999)

In those charts, there are two additional factors directly evaluated from the acknowledge of the temperature at the boundary sections:

$$R = \frac{(T_1 - T_2)}{(t_2 - t_1)}$$

$$S = \frac{(t_2 - t_1)}{(T_1 - t_1)}$$

It is important that FT is not less than 0.75-0.8 as the error would be unacceptable and the chosen heat exchanger configuration would be very inefficient. Furthermore, when reading those charts it is mandatory to be very careful as even a small violation of the first assumption underlying the MTD will invalidate the mathematical derivation, thus resulting in thermodynamically inoperable exchangers. By the way, in some cases FT charts may be employed even for a configuration slightly different from the one they were derived for, without forgetting to be extremely attentive. Finally, it is remarkable that heat exchangers provided by baffles potentially suffer of thermal and physical leakages that are not properly incorporated into correction factor charts, causing less accurate predictions than expectations.

3.2 Cost correlations

The global demand of heat exchangers was economically estimated at 42.7 billion US$ in 2012 and has an annual growth of about 7.8%, so that it was 57.9 billion US$ in 2016 and the market value is expected to be approximatively 78.16 billion US$ in 2020. In particular, the most diffused types are the tubular heat exchanger and the plate one, due to their effective mechanism, low complexity and easier design.

There are some analytical methods which allow the modeler or the constructor to have an idea of what will be the total cost of a heat exchanger, including both capital and operational expense. In effect, they give a good approximation basing on some expressions, factors and coefficients already available in literature.

Marshall and Swift cost index

In this work, we implemented the well-known Marshall and Swift cost index, whose functioning is quite simple. As a matter of fact, most of the available cost data related to whatever equipment refer to past conditions, but the actual prices continuously change over time because of mutations in economic and political conditions. That's why some methods, such as cost indexes, help modelers to update their knowledge of past costs in order to obtain new market prices, which are representative of conditions at a later time or at the present, so that it is possible to attain reliable estimations of future expense to face. Therefore, a cost index is defined as an index value showing the cost of the desired equipment in a given period and relative to a certain base time. The general mathematical expression is (Max S. Peters, 1991):

$$Present\ Cost = Original\ Cost\left(\frac{Present\ Cost\ Index}{Original\ Cost\ Index}\right)$$

Its meaning is quite simple: if the cost at some period in the past is known, the equivalent present one is equal to the original cost multiplied by the ratio of the index value relative to present and the one related to the past. Cost indexes like this give a general estimate, but none of them can take into account every factor, such as local conditions and special technological advancements. Normally, these indexes are quite accurate over periods no longer than 10 years and they are regularly published. A lot of different type of cost indexes exist in literature and they are different from each other, basing on their specific use: some of them allow estimation of equipment costs, others apply specifically to labor, construction, materials or even to more specialized fields. Among those, Marshall and Swift cost index is one of the most common for all-industry and process-industry equipment, but there are also the Nelson-Farrar refinery construction index, the engineering News-Record construction index

and the Chemical Engineering plant cost index. An example of cost indexes listed over a 15-year period is provided in Table 5.

***Table 5**. Different cost indexes from 1995 to 2012 (2014)*

	Marshall and Swift Equipment Cost Index		**Nelson-Farrar Refinery (inflation) Index**	**Chemical Engineering Plant Cost Index (CEPCI)**
Year	*All Industries*	*Process Industry*		
1995	1027.5	1029.0	1392.1	381.1
1996	1039.2	1048.5	1418.9	381.7
1997	1056.8	1063.7	1449.2	386.5
1998	1061.9	1077.1	1477.6	389.5
1999	1068.3	1081.9	1497.2	390.6
2000	1089.0	1097.7	1542.7	394.1
2001	1093.9	1106.9	1579.7	394.3
2002	1104.2	1116.9	1642.2	395.6
2003	1123.6		1710.4	402.0
2004	1178.5		1833.6	444.2
2005	1244.5		1918.8	468.2
2006	1302.3		2008.1	499.6
2007	1373.3		2251.4	525.4
2008	1449.3		n.a.	575.4
2009	1468.6		2217.7	521.9
2010	1457.4		2337.6	550.8
2011			2435.6	585.7
2012				584.6

The Marshall and Swift equipment index, formerly known as Marshall and Stevens, is divided into two classes. The first is the all-industry equipment index, which is basically the arithmetic average of individual indexes for 47 different types of industrial, commercial and housing equipment. The latter is the process-industry equipment index, which is a weighted average of 8 types of different equipment index, whose weights are evaluated regarding on the total product value on various process industries. Moreover, the Marshall and Swift index refers to 100 for the year 1926 as base value from which all the successive cost predictions are extrapolated. Globally, it includes costs of machinery and major equipment plus the expense for installation, fixtures, tools, office furniture and other secondary equipment, thus all the capital expense, also called CAPEX (Max S. Peters, 1991).

So, the final form of the correlations used to estimate the installed cost of the heat exchanger networks modeled in this work will be (Guthrie, Capitol Cost Estimating, Evaluation and Control, 1974):

$$C.I. = \left(\frac{M\&S}{280}\right) \cdot 101.3 \cdot A^{0.65} \cdot (2.29 + F_c)$$

where A is the heat transfer area expressed in [ft2] and the M&S factor will be 1454.5, which refers to the year 2014, while the cost index at the denominator is the base index of the year 1938. Of course, the installed cost includes both purchased and installation cost. The factor F_c is equal to:

$$F_c = (F_d + F_p) \cdot F_m$$

The factors in this expression can be taken from the following tables:

***Table 6**. Material factor F_m relative to both shell and tube side.*

Material $\frac{Shell}{Tube}$	$\frac{CS}{CS}$	$\frac{CS}{Brass}$	$\frac{CS}{Mo}$	$\frac{CS}{SS}$	$\frac{SS}{CS}$	$\frac{CS}{Monel}$	$\frac{Monel}{Monel}$	$\frac{CS}{Ti}$	$\frac{Ti}{Ti}$
F_m	1	1.3	2.15	2.81	3.75	3.1	4.25	8.95	13.05

***Table 7**. Pressure factor F_m depending on the pressure drops along heat exchanger.*

Pressure [psi]	≤ 150	300	400	800	1000
F_p	0	0.1	0.25	0.52	0.55

***Table 8**. Type factor F_m depending on which configuration of heat exchanger is chosen.*

Heat exchanger type	Kettle	Floating Head	U-tube	Fixed tube
F_d	1.35	1.00	0.85	0.8

in which the abbreviations mean: Mo = Molybdenum; Ti = titanium; CS = Carbon steel; SS = Stainless steel.

Furthermore, operative costs must be considered too, as they weight a lot on the total costs of the equipment during its functioning. In fact, they could be a great expense over a certain period of lifetime of the plant and what is more significant is that they are a constant exit of money which

cannot be avoided at all, but only limited and controlled as much as possible. In fact, operative expense, also called OPEX, are subject to oscillations depending on the market trends relative to energy, material supply and fees.

For the case studied in this work, we considered the costs of the utilities as likely average values, thus as constant: the cost of medium pressure steam (30 [bar]) acts as hot utility and its cost is 1.65[€]/1000[lb], while the cooling water acts as cold utility and it has a cost of 0.06[€]/1000[gal]. Since we deal with a heat exchanger, the costs related to the flows are not considered as it has no meaning: indeed, no reaction is assumed to occur during the heat transfer, so what enters the system is exactly what exit it.

4 CASE STUDY

4.1 Single counter-current heat exchanger

In order to develop this work, the first thing to do is to consider a very simple case, that is a single counter-current heat exchanger, with the bypass at both hot and cold side, as shown in the scheme in Figure 20:

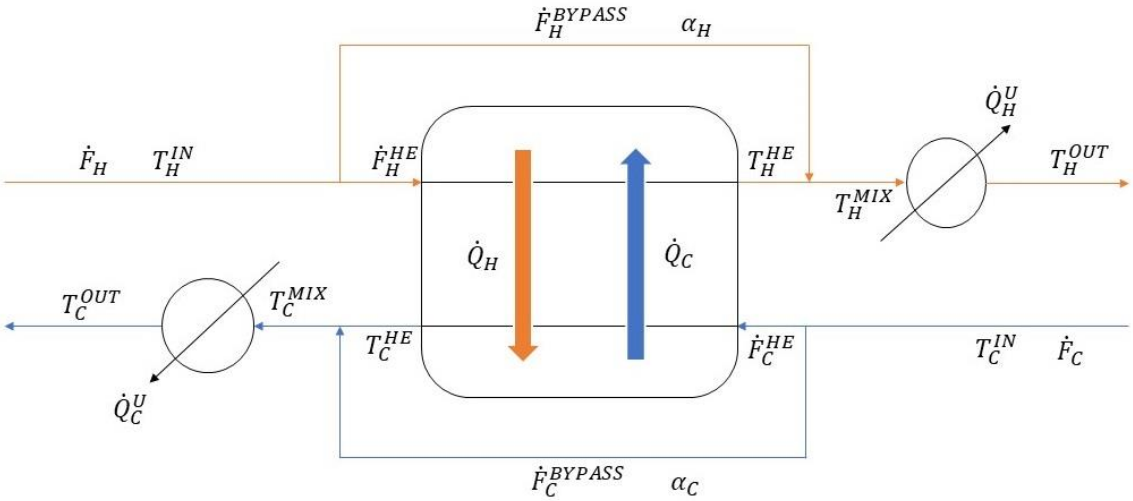


***Figure 19**. Scheme of a single counter-current heat exchanger.*

Therefore, the whole unit is composed of:

- one heat exchanger, in which the heat exchange physically occurs;
- two counter-current flows (one hot and one cold);
- two bypasses (one at the hot side and the other one at the cold side);
- two splitters (one at each side) allow to control how much fluid enters the relative bypasses;
- some valves help the splitters manage the split of entering flows;
- two utilities after the heat exchanger (one working on the hot flux and the other one on the cold side).

4.1.1 Assumptions

Some important assumptions have applied to this system in order to simplify this study and allow us to concentrate more on the task of this work, that is basically the optimization of a heat exchanger network and the approximation of the results by means of a surrogate model. In fact, this is just the first level of modeling, so it is more convenient to assume a case as more ideal as possible, because more realistic situations may be studied in the future, meaning more difficult computations too.

Therefore, the simplifications implemented are:

a) A single, one pass, counter-current heat exchanger with contact surface and without mixing of the hot and cold fluids have been considered;
b) The heat exchange is assumed to have a perfect efficiency ($\eta = 1$);
c) No pressure drops through pipes and single units have been taken into consideration ($\Delta P = 0\ bar$;
d) No energy loss throughout pipes and during the exchange inside the heat exchanger unit itself;
e) All the specific heats c_{P_i} and all the densities ρ_i of each fluid is assumed to be independent from the temperature.

In effect, some correlations for ΔP, c_{P_i} and ρ_i are available on handbooks, like Perry's Chemical Engineers' Handbook, so that their implementation in a successive study would not be difficult.

4.1.2 Degrees of freedom

However, the first thing that must be analyzed before starting any simulation problem is the degrees of freedom related to the system. Indeed, in total this system results to have five degrees of freedom, related to as many pieces of equipment, such as:

- Heat exchanger itself;
- Hot splitter;
- Cold splitter;
- Hot utility;
- Cold utility.

These degrees of freedom may be easily saturated by the following variables, respectively:

- The heat exchanged between the two fluxes inside the heat exchanger
  - $min(\dot{Q}_{HOT}, \dot{Q}_{COLD}) = \dot{Q}_{EXCH}$
- The split factor of the hot flow α_{HOT};
- The split factor of the cold flow α_{COLD};
- The temperature of the hot flow at the outlet of the entire system T_{HOT}^{OUT};
- The temperature of the cold flow at the outlet of the entire system T_{COLD}^{OUT}.

It is important to know that both split factors α_{HOT} and α_{COLD} have a value between 0 and 1.

Moreover, in the MATLAB® code they have been discretized in 100 points and they defined as row vectors:

$$\alpha = [0: 0.01: 1]$$

where the first and last number are the lowest and the highest limit of the vector, respectively, while the number in the middle represents the step, so that the interval from 0 to 1 results to be divided in 100 parts.

4.1.3 Computations

Actually, the effective independent variables are the two split factors α_{HOT} and α_{COLD}, since T_{HOT}^{OUT} and T_{HOT}^{OUT} are specification directly chosen by the modeler, basing on the final target that must be reached; thus, they are arbitrarily implemented into the software. In fact, all the computational efforts relative the thermodynamics is included in two concentric FOR-cycles, one concerning the variation of α_{HOT} and the other one concerning the variation of α_{COLD}. Instead, $\dot{Q}_{EXCH}$ is evaluated by considering the total heat brought by the flows through few simple steps. First, the heat of the flows is calculated by taking the inlet temperature and the final outlet one:

$$\dot{Q}_{HOT}^{HE} = abs\left(\dot{F}_H^{HE} c_P^H (T_H^{OUT} - T_H^{IN})\right)$$
$$\dot{Q}_{COLD}^{HE} = abs\left(\dot{F}_C^{HE} c_P^C (T_C^{OUT} - T_C^{IN})\right)$$

Of course, the flows are exactly the ones entering the heat exchanger itself, after having passed the split valve. Afterward, the lowest heat flux among these is the exchanged heat on which all the successive calculations are based. As in general the heat flux $\dot{Q}$ varies with the actual flow $\dot{F}$, in the MATLAB® code there is an IF-cycle which discretizes between the cases:

A. $\dot{Q}_{COLD}^{HE} < \dot{Q}_{HOT}^{HE}$, meaning that the actual exchanged heat is $\dot{Q}_{EXCH} = \dot{Q}_{COLD}^{HE}$;
B. $\dot{Q}_{COLD}^{HE} \geq \dot{Q}_{HOT}^{HE}$, giving as actual exchanged heat $\dot{Q}_{EXCH} = \dot{Q}_{HOT}^{HE}$.

It is remarkable that the absolute value is considered for the heat fluxes in order to avoid unfeasible quantities. In fact, it doesn't matter if $\dot{Q}$ is positive or negative, that is in one direction or the opposite one, but it is significant to know the amount of heat only. This shrewdness allows us to always attain positive value for all the areas during the successive steps. The general direction of the heat exchange is already accounted for in the IF-cycle structure itself.

Consequently, all the primary computations can be developed inside the IF-cycle. The first thing to evaluate is the temperature obtained right after the exchange has occurred T^{HE} for both hot and cold flow:

$$T_H^{HE} = T_H^{IN} + \frac{\dot{Q}_{EXCH}}{F_H^{HE} c_P^H}$$

$$T_C^{HE} = T_C^{IN} + \frac{\dot{Q}_{EXCH}}{F_C^{HE} c_P^C}$$

Then, before estimating the area of the heat exchanger alone A^{HE}, it is mandatory to calculate the logarithmic temperature difference through the heat exchanger ΔT_{lm} for a counter-current configuration. Thus, the following expressions are used:

$$\Delta T_{lm}^{HE} = \frac{\Delta T_{left} - \Delta T_{right}}{ln \frac{\Delta T_{left}}{\Delta T_{right}}}$$

with

$$\Delta T_{left} = T_{HOT}^{left} - T_{COLD}^{left} = T_H^{IN} - T_C^{HE}$$

$$\Delta T_{right} = T_{HOT}^{right} - T_{COLD}^{right} = T_H^{HE} - T_C^{IN}$$

Now it is possible to obtain the exchange area A^{HE} through the inverse expression reported below:

$$\dot{Q}_{EXCH} = U \cdot A^{HE} \cdot \Delta T_{lm}^{HE} \longrightarrow A^{HE} = \frac{\dot{Q}_{EXCH}}{U \Delta T_{lm}^{HE}}$$

This result as well as the areas of the utilities in the next steps are very important since the final costs of the unit strictly depend on it, as it will be shown later in this work.

At this point, there two different flows at both hot and cold side: the one effectively entering the heat exchanger $\dot{F}_i^{HE}$ and the bypass $\dot{F}_i^{BYPASS}$. In both cases, while the first flow has changed its temperature from T_i^{IN} to T_i^{HE}, the latter one has kept fixed its temperature T_i^{IN}. After having passed the heat exchanger unit, those two fluxes mix each other, so that a mixing temperature T^{MIX} must be evaluated in order to know the real temperature at both outlets and to decide if some utilities are necessary to reach the specifications imposed at the beginning:

$$\dot{F}_i^{HE} C_{p_i} T_i^{HE} + \dot{F}_i^{BYPASS} C_{p_i} T_i^{BYPASS} = \dot{F}_i^{HE} C_{p_i} T_i^{MIX} + \dot{F}_i^{HE} C_{p_i} T_i^{MIX}$$

The C_{p_i} is assumed to be constant, thus it may be collected at both sides of the equation and taken away ending up with the general expression of the mixing temperature:

$$T_i^{MIX} = \frac{\dot{F}_i^{HE} \cdot T_i^{HE} + \dot{F}_i^{BYPASS} \cdot T_i^{BYPASS}}{\dot{F}_i^{HE} + \dot{F}_i^{BYPASS}}$$

More in detail, for the hot and cold side it simply becomes:

$$T_H^{MIX} = \frac{\dot{F}_H^{HE} \cdot T_H^{HE} + \dot{F}_H^{BYPASS} \cdot T_H^{BYPASS}}{\dot{F}_H^{HE} + \dot{F}_H^{BYPASS}}$$

$$T_C^{MIX} = \frac{\dot{F}_C^{HE} \cdot T_C^{HE} + \dot{F}_C^{BYPASS} \cdot T_C^{BYPASS}}{\dot{F}_C^{HE} + \dot{F}_C^{BYPASS}}$$

Now the utilities for the hot flow and the cold flow need to be modeled and the sequence of steps if very similar to heat exchanger case, but simply with different numbers. Indeed, the logarithmic temperature difference ΔT_{ml}^U along both the hot and cold duty is calculated:

$$\Delta T_{ml_H}^U = \frac{(T_H^{MIX} - T_{COOL}) - (T_H^{OUT} - T_{COOL})}{ln\frac{(T_H^{MIX} - T_{COOL})}{(T_H^{OUT} - T_{COOL})}}$$

$$\Delta T_{ml_C}^U = \frac{(T_{STEAM} - T_C^{MIX}) - (T_{STEAM} - T_C^{OUT})}{ln\frac{(T_{STEAM} - T_C^{MIX})}{(T_{STEAM} - T_C^{OUT})}}$$

The temperature T_{COOL} is the temperature of the cooling medium (generally tap water at 25[°C]) used in the utility of the hot flow as it is assumed that the outlet temperature may be higher than the required specification. Furthermore, T_{STEAM} is the temperature of the heating medium (in this case it is medium-pressure steam at 250[°C]) used in the utility of the cold flow as it is assumed that the outlet temperature exiting the heat exchanger may be lower than the required one. For the sake of simplicity, these two temperatures are considered as constant.

Hence, it is possible to evaluate the exchanged heat $\dot{Q}_i^U$ and the corresponding exchange area of the utility A_i^U for both hot and cold flux by means of the following equations:

$$\dot{Q}_H^U = abs\left(\dot{F}_H c_P^{STEAM}\left(T_H^{MIX} - T_H^{OUT}\right)\right)$$

$$A_H^U = \frac{\dot{Q}_H^U}{U_H^U \Delta T_{ml_H}^U}$$

and

$$\dot{Q}_C^U = abs\left(\dot{F}_C c_P^{STEAM}\left(T_C^{MIX} - T_C^{OUT}\right)\right)$$

$$A_C^U = \frac{\dot{Q}_C^U}{U_C^U \Delta T_{ml_C}^U}$$

As it can be clearly seen, the absolute value is taken again for the same reason already explained above.

Finally, the total exchange area is evaluated by summing the heat exchanger area A^{HE} and the two utility exchange areas A_H^U and A_C^U:

$$A^{TOT} = A^{HE} + A_H^U + A_C^U$$

4.1.4 Cost estimation

At this point, we can start the cost estimation of the whole system and its parts. First, the cost variables are estimated by directly implementing utility heat fluxes $\dot{Q}_H^U$ and $\dot{Q}_C^U$ into cost equations taken from the literature (Guthrie, Capitol Cost Estimating, Evaluation and Control, 1974) (Ulrich, 1984) (Pablo F. Navarrete, 2001):

$$CV_{COOL} = 7.78 \cdot 10^{-6} \cdot 8150 * \dot{Q}_H^U \ [\$/y]$$

$$CV_{STEAM} = 0.354 \cdot 10^{-6} \cdot 8150 * \dot{Q}_C^U \ [\$/y]$$

It is mandatory to apply to convert US \$ into €, as the successive correlations work in €, thus the actual value change $value_{change} = 0.88 \ €/\$$ is applied to the cost variables above (d'Italia, 2019). Furthermore, the unit of measure of time must become seconds for the same reason, assuming that the total working hours for one year is about $8000[h]$.

Therefore, the corresponding costs of utilities are:

$$cost_H^U = \left(CV_{COOL}[\$/y] \cdot value_{change}[€/\$]\right)/(8000[h/y] \cdot 60[min/h] \cdot 60[s/min])$$
$$cost_C^U = \left(CV_{STEAM}[\$/y] \cdot value_{change}[€/\$]\right)/(8000[h/y] \cdot 60[min/h] \cdot 60[s/min])$$

The successive step is the estimation of costs relative to purchase and installation of the equipment; actually, the procedure is extremely simple, the necessary factors are chosen among the ones previously shown in the tables and the main expression is:

$$CI^{HE} = \left(\frac{M\&S}{280}\right) \cdot 101.3 \cdot {A^{HE}}^{0.65} \cdot (2.29 + F_c)$$
$$CI_H^U = \left(\frac{M\&S}{280}\right) \cdot 101.3 \cdot {A_H^U}^{0.65} \cdot (2.29 + F_c)$$
$$CI_C^U = \left(\frac{M\&S}{280}\right) \cdot 101.3 \cdot {A_C^U}^{0.65} \cdot (2.29 + F_c)$$

whose unit of measure is [€]. Anyway, it is important to implement this expression to all three areas A^{HE}, A_H^U and A_C^U separately, as it is a non-linear relation. Afterward they may be summed altogether obtaining the total investment cost:

$$CI^{TOT} = CI^{HE} + CI_H^U + CI_C^U$$

The parameters used are the following:

- $M\&S = 1457.4$ (Marshall and Swift cost index), referring to cost of chemical industry in 2014;
- $F_m = 1$, which means that the studied heat exchanger is made of carbon steel at both shell side and tube side $\left(\frac{CS}{CS}\right)$;
- $F_P = 0$, meaning that the pressure drops along the heat exchanger equipment is lower than 150 psi, so the pressure doesn't interfere a lot with the computations;
- $F_d = 0.8$, that is relative to a fixed tube type of heat exchanger.

From these last results it is possible to extrapolate the CAPEX (CAPital EXpense) of this single heat exchanger relative to the whole system or just to all its parts individually, whose graphs are presented later.

The quantities CI^i are all divided by the average lifetime of a common heat exchanger, that is assumed to be 10 years for this case:

$$CAPEX^i = \frac{CI^i}{years}$$

where index $i = HE, U_H, U_C$, which indicates all the different subunits of the system, that is the heat exchanger itself, the hot utility or the cold utility, respectively.
In addition, the OPEX (OPerative EXpense) can be estimated for all the pieces of equipment together or for just all the parts singularly. In practice, these costs are strictly related to the hot and cold duties, which may be considered separately or altogether. As a matter of fact, to evaluate the OPEX it is enough to multiply $cost_H^U$ and $cost_C^U$ (given in $[€/s]$) by the total number of seconds in one year:

$$OPEX^i[€/y] = cost^i[€/s] \cdot 8000[h/y] \cdot 60[min/h] \cdot 60[s/min]$$

At last, the total cost may be evaluated, including both CAPEX and OPEX by simply summing both:

$$COST^i = CAPEX^i + OPEX^i$$

All the costs of every single subunit must be summed altogether in order to attain the global cost of the entire heat exchanger system and the same can be done for global CAPEX and global OPEX:

$$COST^{TOT} = \sum_i COST^i$$

$$CAPEX^{TOT} = \sum_i CAPEX^i$$

$$OPEX^{TOT} = \sum_i OPEX^i$$

where index $i = HE, U_H, U_C$, as already said above
All the results obtained from this MATLAB® code are presented as graphs in the next section.

4.1.5 Surrogate modeling for case study

The final step is actually the main target of this case study and it is the surrogate model construction by using the new aforesaid software ALAMO. So, some training points are taken from the results obtained with MATLAB®, then they are properly implemented in ALAMO, so that it generates a surrogate model by means of regression.

First of all, the data have been chosen by applying three common sampling techniques, such Latin Hypercube Sampling (LHS), Cluster Sampling (CS) and Systematic Sampling (SS). For the first one it is possible to use the MATLAB® command $lhsdesign\ (n, p)$, where n is the number of desired training points and p is the number of variables, in this case they are just two (α_H and α_C). For the other two techniques, a direct sampling from the workplace of MATLAB® was made as its application is easier that LHS. In particular, a simple $m \times p$ grid of training points is arbitrarily generated for the SS; instead, some clusters, that is some groups, of training points are chosen on the surfaces and among the results in MATLAB® workplace for the CS. The strategies used to choose the training points in the three sampling techniques are the following:

- In LHS, the MATLAB® command $lhsdesign$ above mentioned has just been implemented and it has given back the desired points;
- In SS, a well-defined grid was overlapped on the table of results in the MATLAB® workplace;
- In CS, some clusters of points are chosen in those regions where the surface has more critic behavior, such as maximum or minimum points, saddle points, asymptotes, etc.

After having sampled, the data points are implemented in ALAMO (in the blue box), which requires to know the number of input variables, the number of output variables, the number of points to work on (see red box) and to indicate the name of variables, the lowest and the highest value for each independent variable (see green box); the view of this step is reported in figure 21.

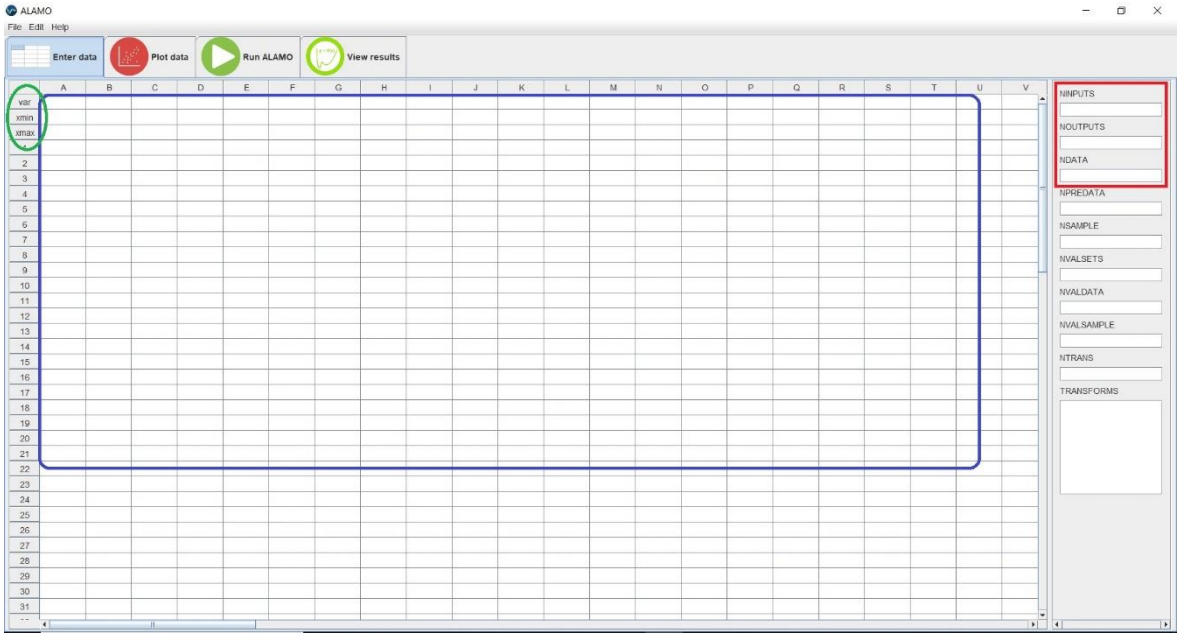


***Figure 20**. First window 'Enter data' of ALAMO.*

As ALAMO has acquired the inputs, it is available an overview chart, which is basically a cartesian graph where it is possible to select the variables on both x and y axis. At this point, the user can observe the behavior of the inserted points as they have been provided, before any regression occurs, as it is in Figure 22.

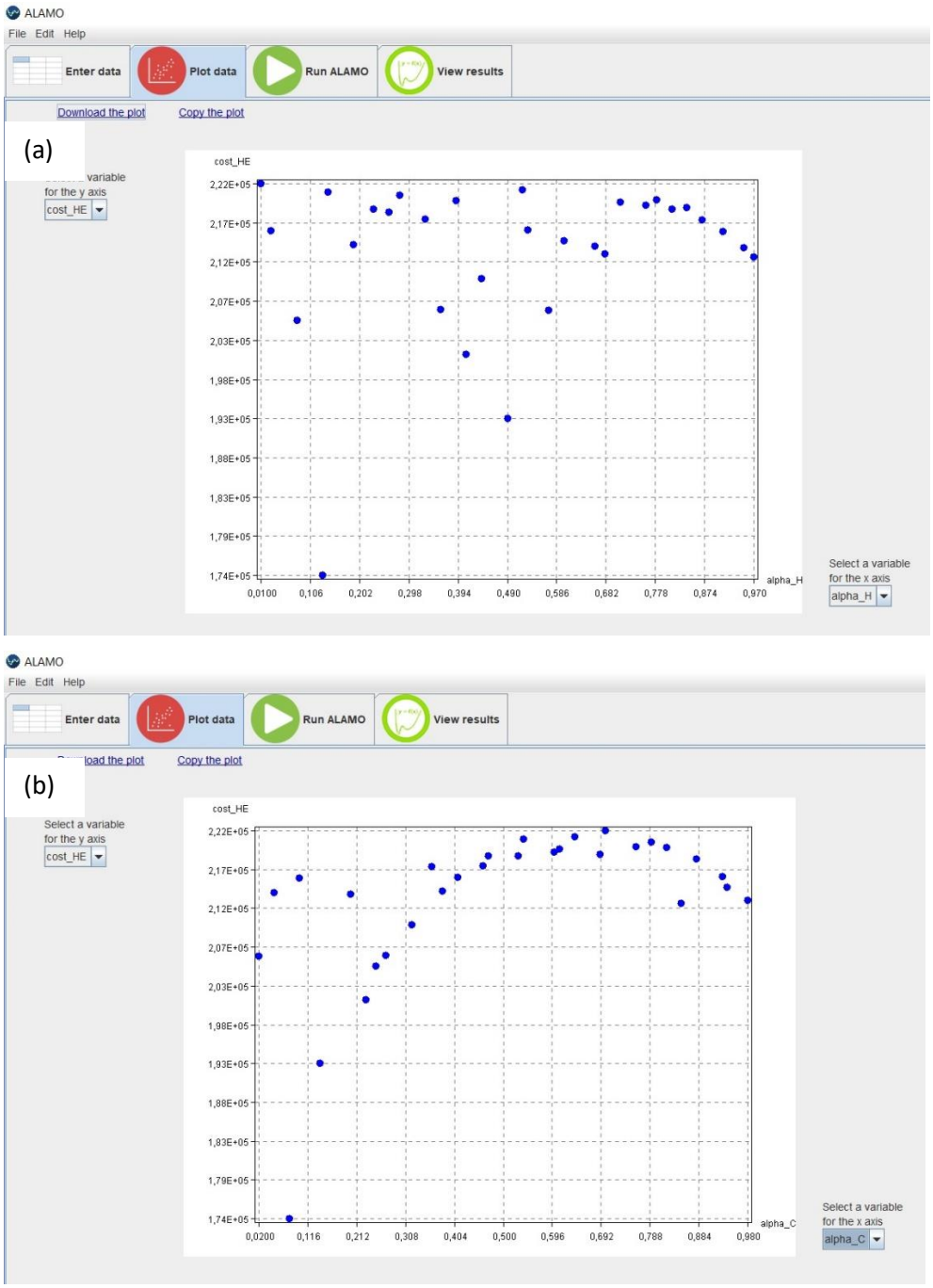


Figure 21. *Second window 'Plot data' of ALAMO. (a)* α_H *on the x-axis and* $Total\ cost$ *on the y-axis. (b)* α_C *on the x-axis and* $Total\ cost$ *on the y-axis.*

Afterward, at this fundamental step it is possible to set the regression which ALAMO will process; hence, it is a sort of control panel of all surrogate modeling operation, whose screen view is reported in Figure 23.

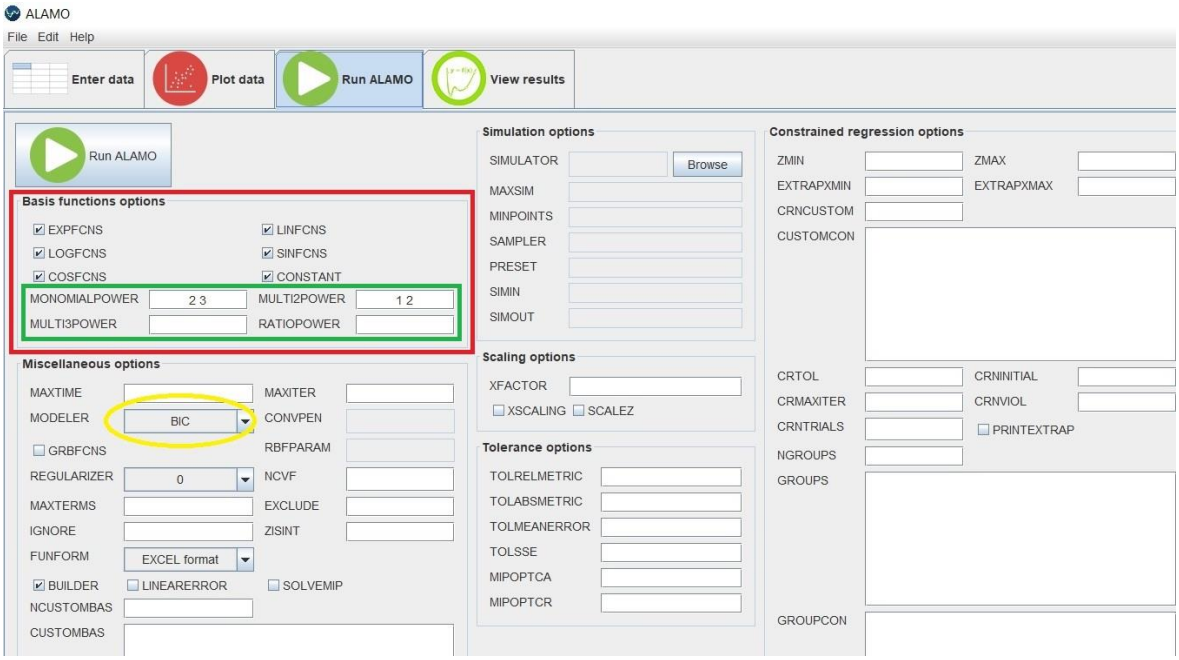


***Figure 22**. Third window 'Run ALAMO' of ALAMO.*

I. In the red box, a list of basis function is available and basing on the nature of the curves or surfaces the results present we can select the most fitting ones;

II. In the green box, we can give ALAMO the possibility to exploit some polynomial functions for the regression; for example, there may be monomial powers of the single variables ($MONOMIALPOWER$) or binomial powers ($MULTI2POWER$), where two variables are combined together. This information must be provided as row vector;

III. In the yellow circle, there is the choice of the model fitness measure, in this case BIC has been chosen, as it is largely used;

IV. Simulation options and scaling options guide the simulation applied by ALAMO, but the default one (BARON, (Sahinidis N. V., 2015)) is enough for our case;

V. Tolerance options are available in order to manage the accuracy of the software, but the default ones have been chosen, as our successive regressions have demonstrated to be accurate enough;

VI. Constrained regression options are other possible instruments to characterize the regression, but our target is just to prove if ALAMO has good performances for simple regression and consecutive surrogate modeling.

VII. Miscellaneous options allow the user to manage the regression in different ways, for example imposing the maximum time of computations ($MAXTIME$), the maximum number of iterations ($MAXITER$), the $REGULARIZER$ (which helps to obtain surrogate points very close to the training ones, but it increases a lot the computational requirements and it is not necessary for our simple case), etc.

In the final step, ALAMO shows all the results, as it is reported in Figure 24.

(a)

(b)

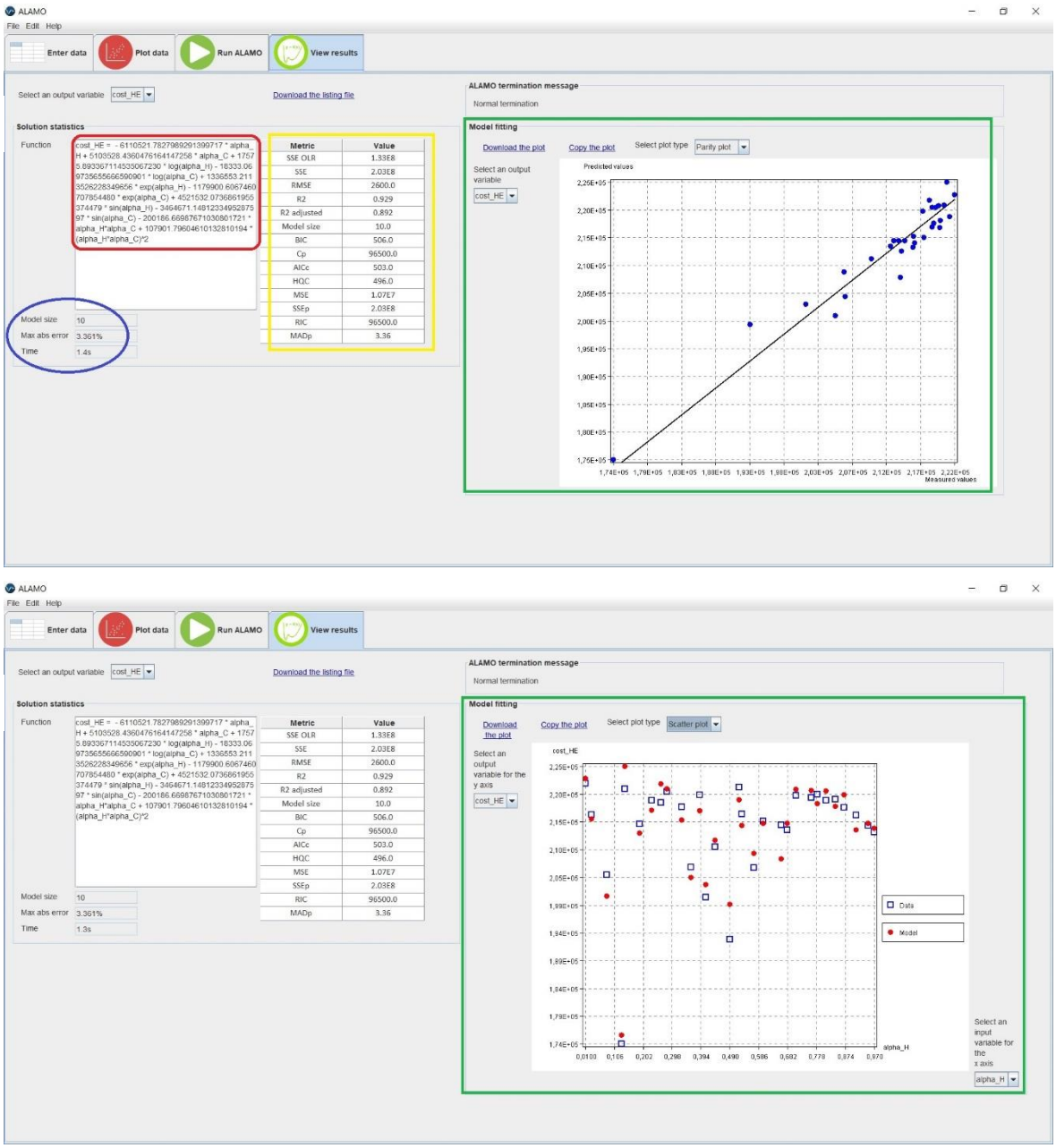


***Figure 23**. Fourth window 'View results' of ALAMO. (a) 'Parity plot' is shown. (b) 'Scattered plot' is shown.*

As it can be seen, the analytical function in the red circle is the objective of this work, as it is the result of the regression which ALAMO has applied to the training sets coming from the original MATLAB® code. It should approximate as best as possible the behavior of the previous surfaces in order to describe any phenomenon, so that successive works and elaborations will be able to exploit that mathematical function and many operations will be way easier than working directly on the simulation itself, like integration, derivation or other transformations. The table in the yellow box

reports the most common model fitness metrics, even if the one ALAMO used in our case is the BIC, as already said above. Below the function the software reports the model size, the maximum absolute error and the time spent for computations (see blue circle). Finally, two types of graph are available in the figure on the right side of the screen (inside the green boxes in the previous two figures):

i. The parity plot, which shows how much distant the surrogate points are from the average value, represented by the black obliquus line;
ii. The scattered plot, which is a normal cartesian graph where the user can select the variable on the x and y axis in order to make a comparison among them. In addition, it shows both the data points (blue ones) and the model points (red ones), so that the user can observe the accuracy of the surrogate.

After ALAMO has provided a surrogate model related to the previous simulation, next step is to implement it in MATLAB® again in order to ascertain its accuracy and reliability, by simply copying the surrogate function and adapting it to the MATLAB® code, thus obtaining:

$$COST^{TOT}_{ALAMO\,LHS} = [function\ from\ ALAMO]$$
$$COST^{TOT}_{ALAMO\,SS} = [function\ from\ ALAMO]$$
$$COST^{TOT}_{ALAMO\,CS} = [function\ from\ ALAMO]$$

This procedure is repeated for every aforesaid sampling technique and for different amounts of initial training points. In particular, we arbitrary chose the following samples:

- For LHS, 50, 30 and 20 points;
- For SS, 6 x 5 grid (30 points), 6 x 4 grid (24 points) and 4 x 4 grid (16 points);
- For CS, 7 clusters of 5 points each one, 6 clusters of 4 points each one and 6 clusters of 3 points each one.

Afterward, it may be extremely useful to have a sensitivity parameter which gives the modeler mathematical measure of reliability of the surrogate model. Therefore, the difference between MATLAB® results and ALAMO function is evaluated:

$$DIFF_{MATLAB-ALAMO} = COST^{TOT} - COST^{TOT}_{ALAMO}$$

Of course, this operation is repeated for all the different sampling techniques cases and all the different samples. Then, sensitivity metrics is estimated for every result by dividing the difference $DIFF_{MATLAB-ALAMO}$ by $COST^{TOT}$, in order to have a non-dimensional measure, which is more interesting to analyze:

$$SENSITIVITY_{MATLAB-ALAMO} = \frac{DIFF_{MATLAB-ALAMO}}{COST^{TOT}}$$

The range of this index is $[0,1]$, thus it is sufficient to multiply by 100 if we want the percentage value. All the graphs obtained until now are reported in the next section.

4.2 Two counter-current heat exchangers

The second level of development of a heat exchanger network, like Grossmann one, is to add another heat exchanger of the same type of the previous one, so that the scheme of the whole system is presented in Figure 25:

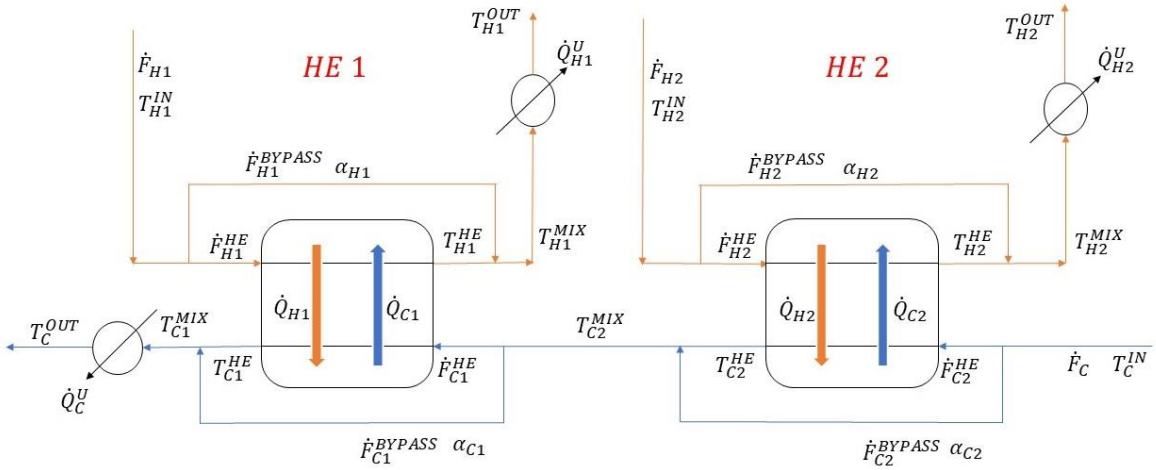


***Figure 24**. Scheme of a network composed of two counter-current heat exchangers.*

Therefore, the whole unit is composed of:

- two heat exchangers, in which the heat exchange physically occurs and connected through the cold counter-current flux;
- three counter-current flows (two hot and one cold);
- four bypasses (for each heat exchanger unit, one is at the hot side and the other one is at the cold side);
- four splitters (one along each hot current and two along the cold flow) allow to control how much fluid enters the relative bypasses;
- some valves help the splitters manage the split of entering flows;
- three utilities after the heat exchangers, at the end of each flow (two working on the two hot fluxes and the other one working on the cold side).

4.2.1 Assumptions

The type of heat exchanger used as well as the assumptions relative to physical and thermodynamic properties are exactly the same of previous case. By doing so, it is easier to adapt the code of a level to the one of the successive levels, as initial data provided to MATLAB® are basically identical and it is sufficient to modify just the FOR- and IF-cycles.

4.2.2 Degrees of freedom

Instead, this second-level system results to have nine degrees of freedom, related to as many pieces of equipment, such as:

- 2 heat exchanges;
- 2 hot splitters;
- 2 cold splitters;
- 2 hot utilities;
- Cold utility.

In a similar way to the previous base case, these degrees of freedom may be easily saturated by the following variables, respectively:

- The heat exchanged between the two fluxes inside the 2 heat exchangers
  - $min(\dot{Q}^1_{HOT}, \dot{Q}^1_{COLD}) = \dot{Q}^1_{EXCH}$
  - $min(\dot{Q}^2_{HOT}, \dot{Q}^2_{COLD}) = \dot{Q}^2_{EXCH}$
- 2 split factors of the hot flows α^1_{HOT} and α^2_{HOT};
- 2 split factors of the only cold flow α^1_{COLD} and α^2_{COLD};
- The temperatures of both hot flows at the outlet of the entire system ${T^{OUT}_{HOT}}^1$ and ${T^{OUT}_{HOT}}^2$;
- The temperature of the cold flow at the outlet of the entire system T^{OUT}_{COLD}.

As before, the effective independent variables are the four split factors α^1_{HOT}, α^2_{HOT}, α^1_{COLD} and α^2_{COLD}, since the outlet temperatures ${T^{OUT}_{HOT}}^1$, ${T^{OUT}_{HOT}}^2$ and T^{OUT}_{COLD} are again specifications directly chosen by the modeler; thus, they are arbitrarily implemented into the software. Instead, $\dot{Q}^1_{EXCH}$ and $\dot{Q}^2_{EXCH}$ are evaluated through the same simple procedure of the previous case.

4.2.3 Computations and cost estimation

Luckily, the computations are quite similar to the single heat exchanger case, therefore it is not necessary to indicate all the implemented equations, as they are equal and it would be useless to this work. However, there are two significant differences from the previous code:

1) There are two IF-cycles, because the discretization between the two cases $\dot{Q}^{HE}_{COLD} < \dot{Q}^{HE}_{HOT}$ and $\dot{Q}^{HE}_{COLD} \geq \dot{Q}^{HE}_{HOT}$ must be done for both heat exchangers;
2) The FOR-cycles are relative to all hot and cold split factors α^1_{HOT} , α^2_{HOT} , α^1_{COLD} and α^2_{COLD}, so they are four and the consecutive computations and structure of the whole code will be unavoidably heavier.

However, the part of code concerning the cost estimation stays almost unchanged too, except for some little adaptations. Furthermore, the results are expressed in four dimensions and this characteristic is directly connected on the number of independent variables, which is exactly four.

4.2.4 Surrogate modeling for two counter-current heat exchangers

Like for thermodynamic and economical calculations, the surrogate modeling by means of ALAMO has exactly the same procedure of the base case.

4.3 Three counter-current heat exchangers

In order to reach the third level of development of the final superstructure, the third heat exchanger has been added to the previous system, with appropriate modifications to the entire system, so that the new configuration is shown in Figure 26:

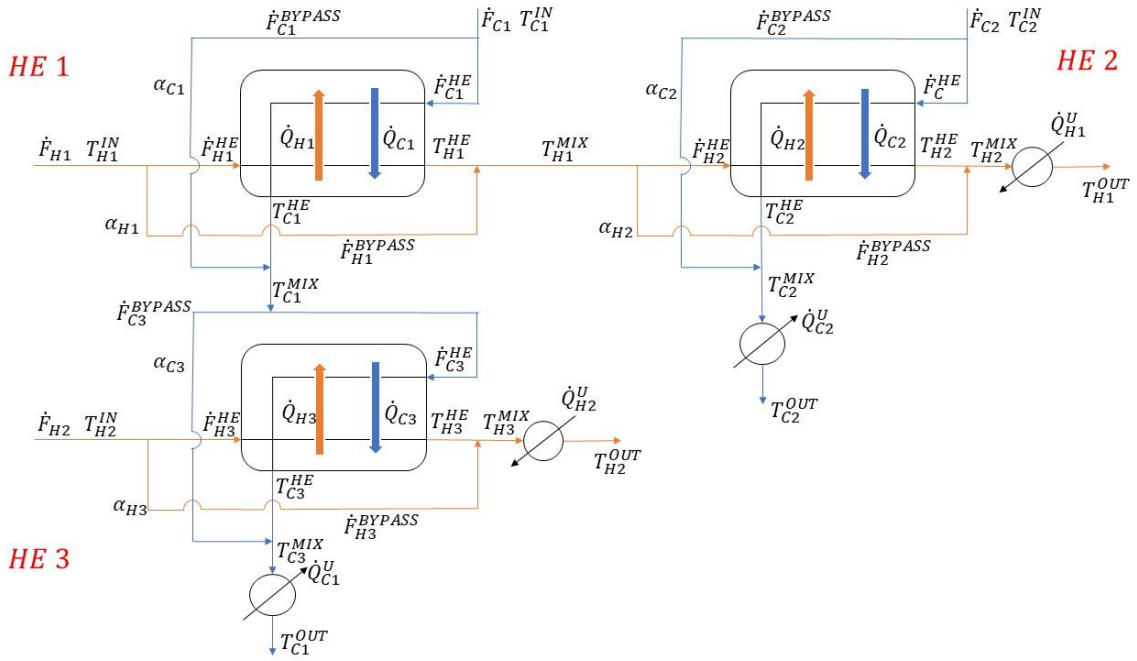


***Figure 25**. Scheme of a network composed of three counter-current heat exchangers.*

So, the whole unit is composed of:

- three heat exchangers, in which the heat exchange physically occurs and connected by means of the hot flow $\dot{F}_{H1}$ and the cold flow $\dot{F}_{C1}$;
- two hot fluxes and two cold fluxes, which flow with counter-current directions among themselves when they enter the heat exchanger units;
- six bypasses (for each heat exchanger unit, one is at the hot side and the other one is at the cold side);
- six splitters (two along $\dot{F}_{H1}$, two along $\dot{F}_{C1}$, one along $\dot{F}_{H2}$, one along $\dot{F}_{C2}$) allow to control how much fluid enters the relative bypasses;
- some valves help the splitters manage the split of entering flows;
- four utilities after the heat exchangers, at the end of each flow (two working on the two hot fluxes and two working on the cold ones).

As it can be clearly seen, the global configuration at this level visually seems to the final superstructure, which is the actual target of this research.

4.3.1 Assumptions

The type of heat exchanger used as well as the assumptions relative to physical and thermodynamic properties are exactly the same of base case study.

4.3.2 Degrees of freedom

In total, this third-level structure has thirteen degrees of freedom, related to as many pieces of equipment, such as:

- 3 heat exchanges;
- 3 hot splitters;
- 3 cold splitters;
- 2 hot utilities;
- 2 cold utility.

In a similar manner of previous cases, these degrees of freedom may be saturated by the following variables, respectively:

- The heat exchanged between the two fluxes inside each heat exchanger
  - $min(\dot{Q}^1_{HOT}, \dot{Q}^1_{COLD}) = \dot{Q}^1_{EXCH}$
  - $min(\dot{Q}^2_{HOT}, \dot{Q}^2_{COLD}) = \dot{Q}^2_{EXCH}$
  - $min(\dot{Q}^3_{HOT}, \dot{Q}^3_{COLD}) = \dot{Q}^3_{EXCH}$
- 3 split factors of the hot flows α^1_{HOT} , α^2_{HOT} and α^3_{HOT};
- 3 split factors of the cold flow α^1_{COLD} , α^2_{COLD} and α^3_{COLD};
- The temperatures of both hot flows at the outlet of the entire system $T^{OUT\,1}_{HOT}$ and $T^{OUT\,2}_{HOT}$;
- The temperatures of both cold flows at the outlet of the entire system $T^{OUT\ 1}_{COLD}$ and $T^{OUT\ 2}_{COLD}$.

Again, the effective independent variables are the six split factors α^1_{HOT}, α^2_{HOT}, α^3_{HOT} , α^1_{COLD} , α^2_{COLD} and α^3_{COLD} , as the outlet temperatures $T^{OUT\,1}_{HOT}$, $T^{OUT\,2}_{HOT}$, $T^{OUT\ 1}_{COLD}$ and $T^{OUT\ 2}_{COLD}$ are specifications defined by the modeler and they are arbitrarily implemented into the software. Instead, $\dot{Q}^1_{EXCH}$, $\dot{Q}^2_{EXCH}$ and $\dot{Q}^3_{EXCH}$ are evaluated through the same procedure of previous cases.

4.3.3 Computations and cost estimation

Even in this case, computations as well as cost estimation are essentially equal to the single heat exchanger case, so the implemented equations will not be shown below for the same reason of before. By the way, there are again two significant differences from the base case study:

1) Now the IF-cycles are three, because the discretization between the two cases $\dot{Q}_{COLD}^{HE} < \dot{Q}_{HOT}^{HE}$ and $\dot{Q}_{COLD}^{HE} \geq \dot{Q}_{HOT}^{HE}$ must regard every heat exchanger;
2) The concentric FOR-cycles relative to all hot and cold split factors α_{HOT}^{1}, α_{HOT}^{2}, α_{HOT}^{3}, α_{COLD}^{1}, α_{COLD}^{2} and α_{COLD}^{3} are six, thus resulting in heavier and heavier computational efforts due to the more complex structure of the code.

In fact, the time spent by the computer for the calculations starts to be more or less relevant with respect to previous two cases (slightly less than one minute), but it is not a big issue. Instead, the main problem is the structure of the results, since MATLAB® must work in six dimensions environment, thus difficult to show and analyze. In effect, since there are six independent variables (the six split factors), MATLAB® automatically generates 6D matrixes and their visual presentation is quite difficult to deal with.

4.4 Four counter-current heat exchangers

This is the final step in the development of heat exchanger network and the fourth heat exchanger is added to the previous configuration, so that it is very similar to the Grossmann superstructure introduced at the beginning of this work. The resulting scheme is presented in Figure 27:

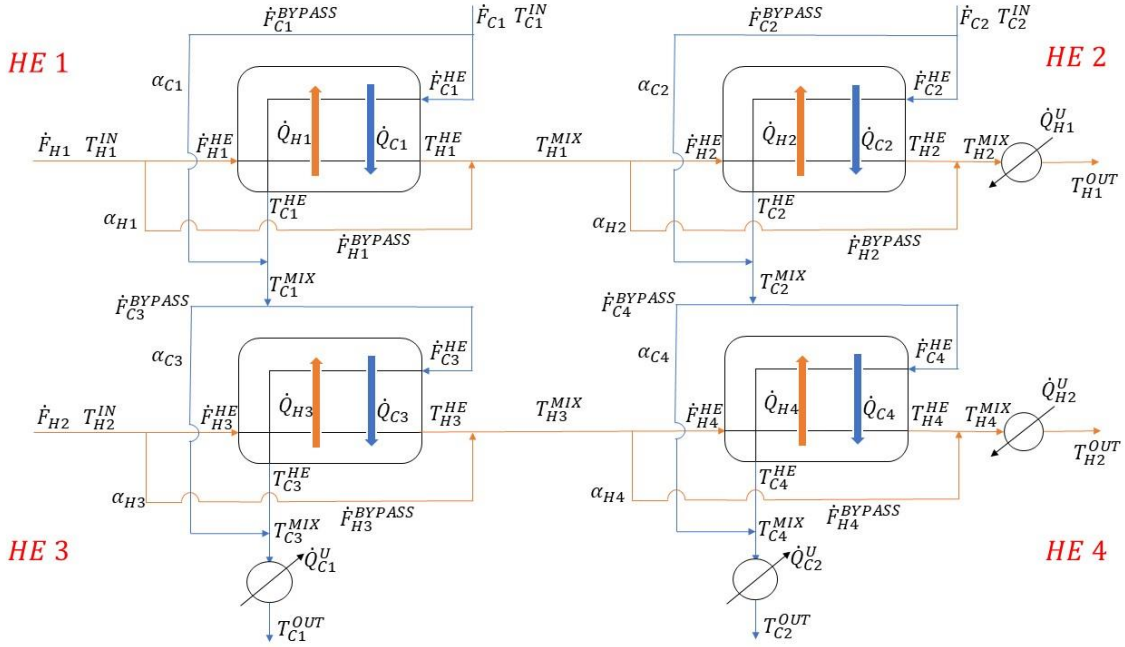


***Figure 26**. Scheme of a network composed of four counter-current heat exchangers (similar to Grossmann HEN).*

So, the main components of this system are:

- four heat exchangers all interconnected among them, in which the heat exchange;
- two hot fluxes and two cold fluxes, which flow with counter-current directions among themselves when they enter the heat exchanger units;
- six bypasses (for each heat exchanger unit, one is at the hot side and the other one is at the cold side);
- eight splitters (two along each current) allow to control how much fluid enters the relative bypasses;
- some valves help the splitters manage the split of entering flows;
- four utilities after the heat exchangers, at the end of each flow (two working on the two hot fluxes and two working on the cold ones).

4.4.1 Assumptions

Again, the type of heat exchanger used as well as the assumptions relative to physical and thermodynamic properties are exactly the same of base case study, as already made for the second and third level of development. In this way, all the changes which may be noted are strictly related to the increasing complexity of the system only, so that the study of this work is more focused on optimization and model surrogating, which are the real targets.

4.4.2 Degrees of freedom

In total, this third-level structure has sixteen degrees of freedom, related to as many pieces of equipment, such as:

- 4 heat exchanges;
- 4 hot splitters;
- 4 cold splitters;
- 2 hot utilities;
- 2 cold utility.

In a similar manner of previous cases, these degrees of freedom may be saturated by the following variables, respectively:

- The heat exchanged between the two fluxes inside each heat exchanger
  - $min(\dot{Q}^1_{HOT}, \dot{Q}^1_{COLD}) = \dot{Q}^1_{EXCH}$
  - $min(\dot{Q}^2_{HOT}, \dot{Q}^2_{COLD}) = \dot{Q}^2_{EXCH}$
  - $min(\dot{Q}^3_{HOT}, \dot{Q}^3_{COLD}) = \dot{Q}^3_{EXCH}$
  - $min(\dot{Q}^4_{HOT}, \dot{Q}^4_{COLD}) = \dot{Q}^4_{EXCH}$
- 4 split factors of the hot flows α^1_{HOT} , α^2_{HOT} , α^3_{HOT} and α^4_{HOT};
- 4 split factors of the cold flow α^1_{COLD} , α^2_{COLD} , α^3_{COLD} and α^4_{COLD};
- The temperatures of both hot flows at the outlet of the entire system $T^{OUT\,1}_{HOT}$ and $T^{OUT\,2}_{HOT}$;
- The temperatures of both cold flows at the outlet of the entire system $T^{OUT\ 1}_{COLD}$ and $T^{OUT\ 2}_{COLD}$.

Again, the effective independent variables are the 8 split factors (α^1_{HOT}, α^2_{HOT}, α^3_{HOT} and α^4_{HOT} for the hot side; α^1_{COLD} , α^2_{COLD} , α^3_{COLD} and α^4_{COLD} for the cold side), as the outlet temperatures $T^{OUT\,1}_{HOT}$, $T^{OUT\,2}_{HOT}$, $T^{OUT\ 1}_{COLD}$ and $T^{OUT\ 2}_{COLD}$ are specifications defined by the modeler and they are arbitrarily

implemented into the software. Instead, $\dot{Q}_{EXCH}^{1}$, $\dot{Q}_{EXCH}^{2}$, $\dot{Q}_{EXCH}^{3}$ and $\dot{Q}_{EXCH}^{4}$ are evaluated through the same procedure of previous cases.

4.4.3 Computations and cost estimation

Finally, computations, equations and cost estimation of this fourth code are almost the same of the ones used inside previous codes, so that they will be not reported. The base case shows the basic equations, as the procedure is basically unchanged, even if it is simply repeated for every heat exchanger with appropriate modifications. Anyway, also in this case there are the same differences from the base case study:

1) The IF-cycles become four (one for each heat exchanger) and their structure is essentially the same of previous codes;
2) The concentric FOR-cycles relative to every single split factor (α_{HOT}^{1} , α_{HOT}^{2}, α_{HOT}^{3}, α_{HOT}^{4}, α_{COLD}^{1}, α_{COLD}^{2}, α_{COLD}^{3} and α_{COLD}^{4}) are eight, thus resulting in heavier and heavier computational efforts due to the more complex structure of the code.

At this level, the time needed to process all the information is relevant for a normal computer: that's why it has been necessary to reduce the discretization of the split factors by one order of magnitude (10 points instead of 100) in order to avoid too long calculations. In addition, the main problem is that the results are in eight dimensions, since the code is based on eight independent variables. Hence, it is very difficult to show and analyze what we obtain, both numbers and graphs.

5 RESULTS

5.1 Single counter-current heat exchanger

5.1.1 Results of MATLAB®

Since the base case with single heat exchanger is the simplest case studied in this work, the results can be expressed through many graphs and their analysis may be very interesting. First of all, it is not sure that the heat is always exchanged from hot to cold side, as its direction depends on the quantity of fluid flowing in the tube, thus on the split factors α_i, which are independent variables (in this simulation, the range [0,1] of α_{HOT} has been divided into 100 points, while the one of α_{COLD} has been divided into 40 points). Hence, the two alternatives are shown below in Figure 28.

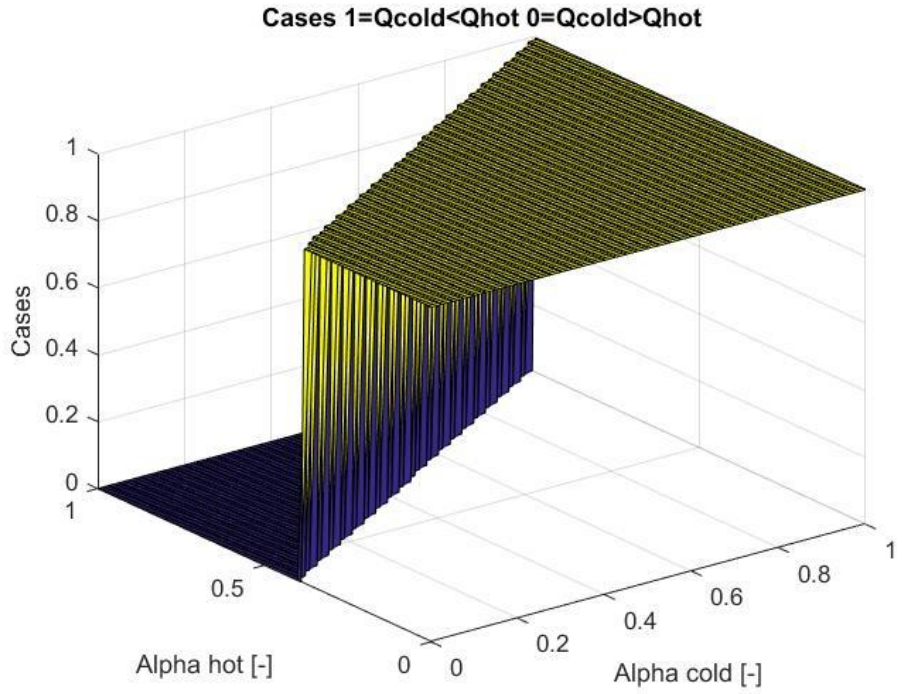
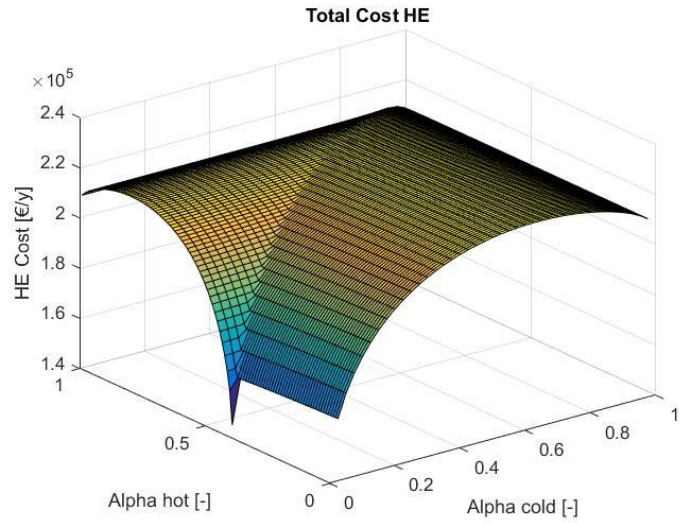


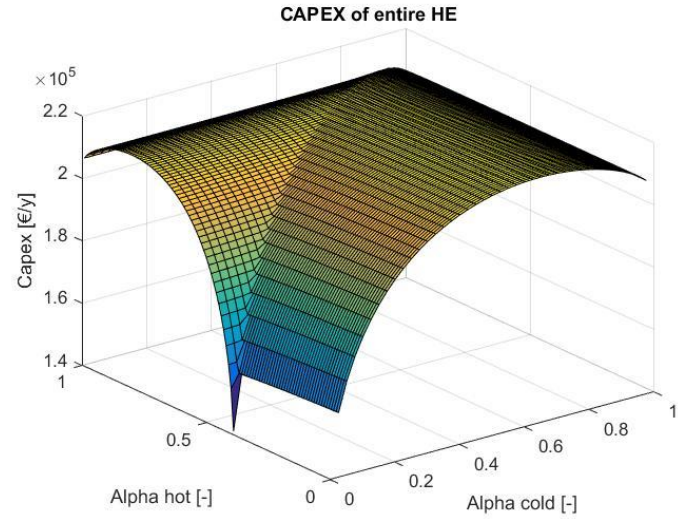
***Figure 27**. Cases of single heat exchanger: case (0) means that cold flow exchanges toward hot flow; case (1) means that hot flow exchanges toward cold flow.*

As it can be clearly seen, it is more likely that the amount of heat inside hot flux $\dot{Q}_{HOT}^{HE}$ is larger than the one in cold flux $\dot{Q}_{COLD}^{HE}$, meaning that heat goes from hot to cold side more easily. Nevertheless, it can even happen the contrary, just because the combination between α_{HOT} and α_{COLD} makes the amount of cold flow bigger than the one of hot flow as well as the amount of energy brought inside. Considering the first target of this work, that is the optimization, the most relevant graph is the total cost of the single heat exchanger and the utilities together, thus including both installation costs and operative expense, but also the graphs of CAPEX and OPEX separately.

(a)



(b)



(c)

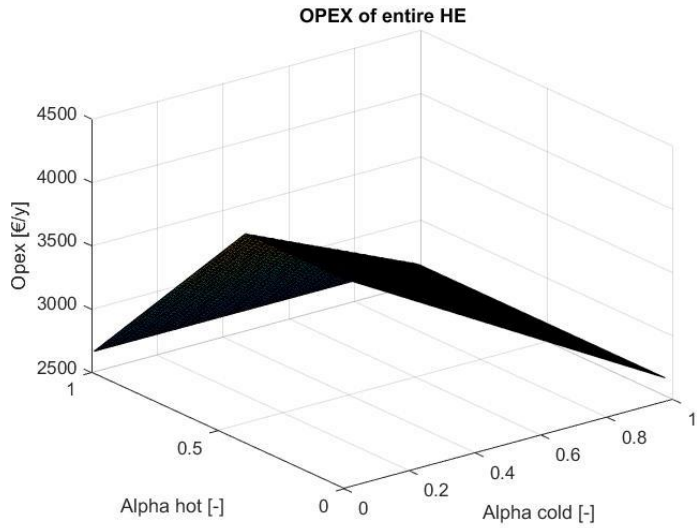


Figure 28*. (a) Total cost of the entire heat exchanger system. (b) CAPEX of the entire heat exchanger system. (c) OPEX of the entire heat exchanger system.*

The first feature which can be observed is that total costs and CAPEX have the same shape and almost equal quantities, since installation and purchase expense have much more weight than the OPEX: this behavior probably depends on the assumptions and the correlations we used for a heat exchanger and on utility costs we chose. Actually, the only difference is that total costs are slightly larger than CAPEX and they grow as both α_{HOT} and α_{COLD} increase. Furthermore, in all three graphs the division between the case where hot fluid transfers to cold fluid and the opposite one is clear. However, the main difference between OPEX and CAPEX surfaces is that as the first decreases, the latter increases, so they are more or less inversely proportional. Finally, the maximum total costs of the whole system is when at least α_{HOT} or α_{COLD} is about 0.8.

Similar features are observable in the graphs of total costs of the single heat exchanger alone, the hot duty and the cold duty.

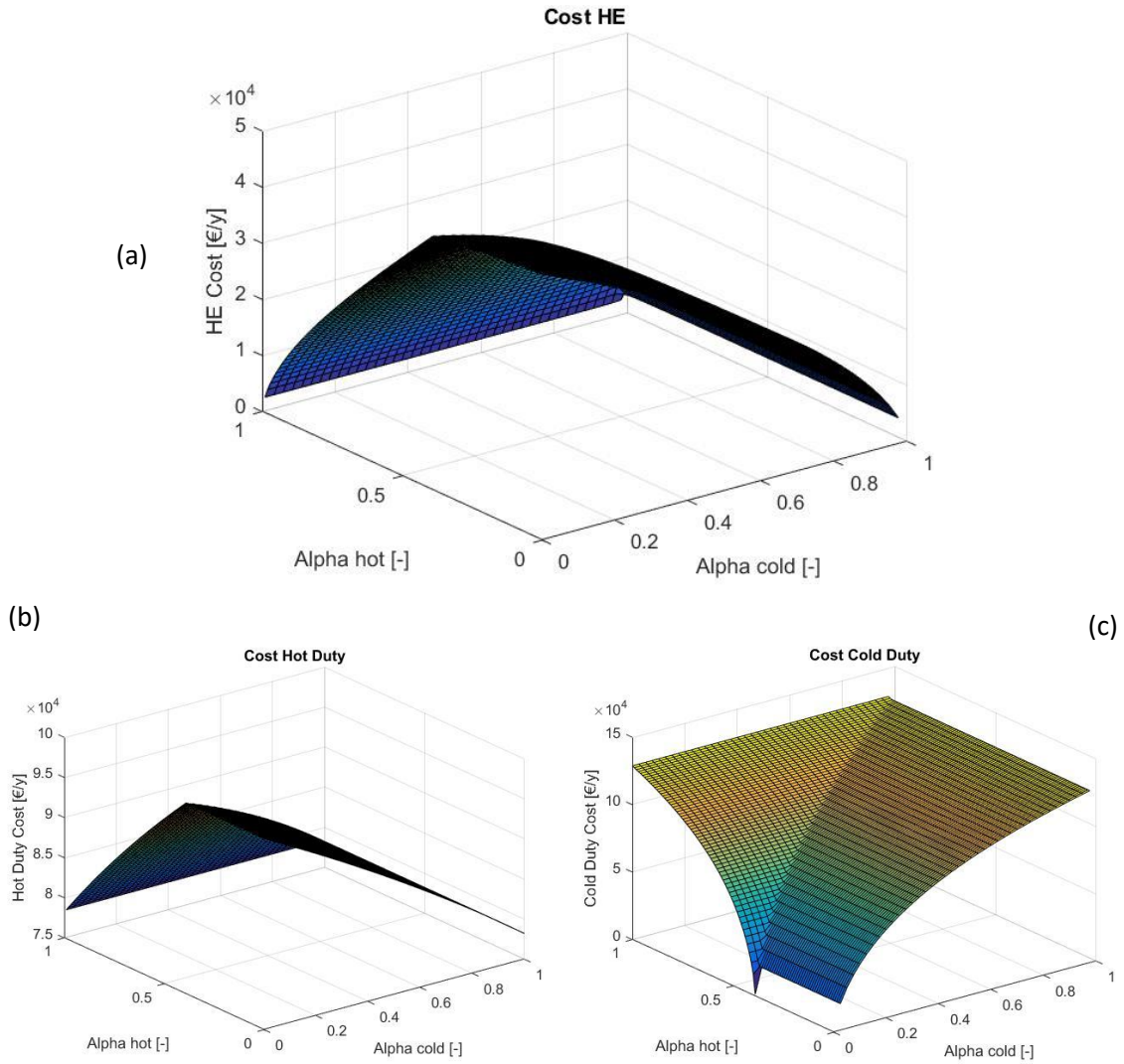


Figure 29*. (a) Total cost relative to single heat exchanger unit. (b) Total cost of hot duty. (c) Total cost of hot duty.*

As before, the division between the two cases is very net, but there are some differences with respect to previous figures. It is observable that heat exchanger cost and hot duty cost have basically the same behavior, but the first is almost one order of magnitude larger than the second one, because the only cost for the heat exchanger alone is just the exchange area, while both hot and cold utilities have to include also the expense for cooling water and heating steam. The surface of cold duty cost presents an increasing slope as α_{HOT} and α_{COLD}, thus presenting its maximum along the perimeter where at least one of two split factors is 1. Instead, the maximum of in the other two graphs is when α_{COLD} and α_{HOT} has a value between 0 and 0.4.

Next graphs at Figure 31 show the results relative to the total exchange area of the entire system, thus including heat exchanger, hot and cold utilities, and each single unit separately.

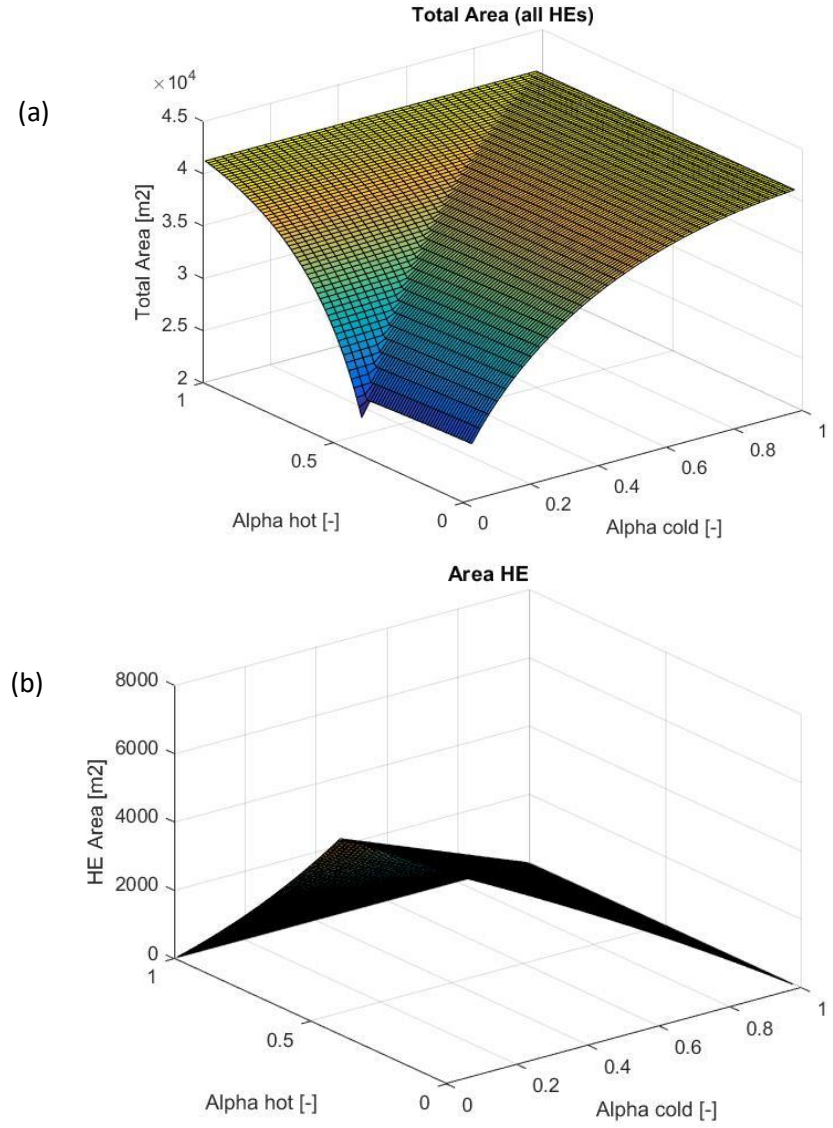
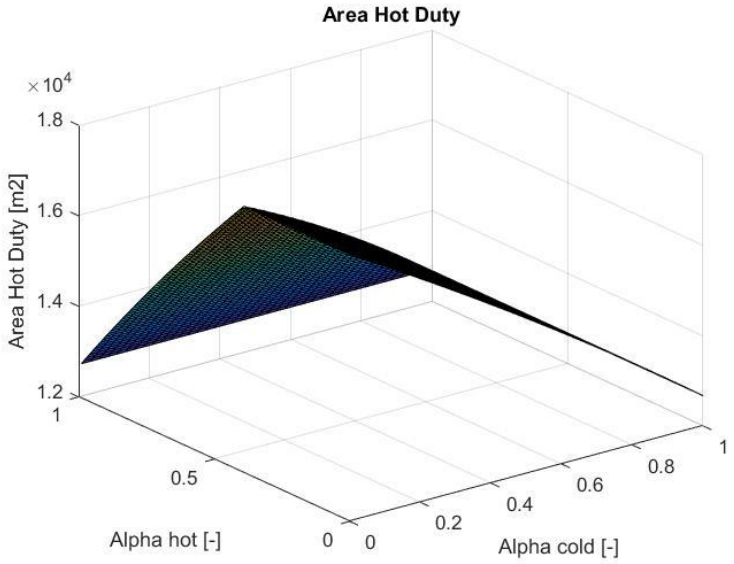


Figure 30*. (a) Total heat exchange area relative to all units together (heat exchanger itself, hot duty and cold duty). (b) Exchange area of heat exchanger only.*

It is immediately clear the strict dependence between cost and area: in effect, they have a very similar behavior for every single unit. Anyway, the maximum for the heat exchanger only is again for $\alpha_{HOT} = [0,0.4]$ and $\alpha_{COLD} = 0$, like before, while the dependence area-split factors is more linear than cost-split factors, likely due to the nonlinearity of cost correlations. Instead, in the case of the total exchange area there is the same behavior of total cost, but the maximum is shifted to the edge of the surface, thus for those points which are related at least to one of the two split factors equal to 1.

Finally, the other two areas are presented in Figure 32, that is area of hot and cold utility separately.

(a)



(b)

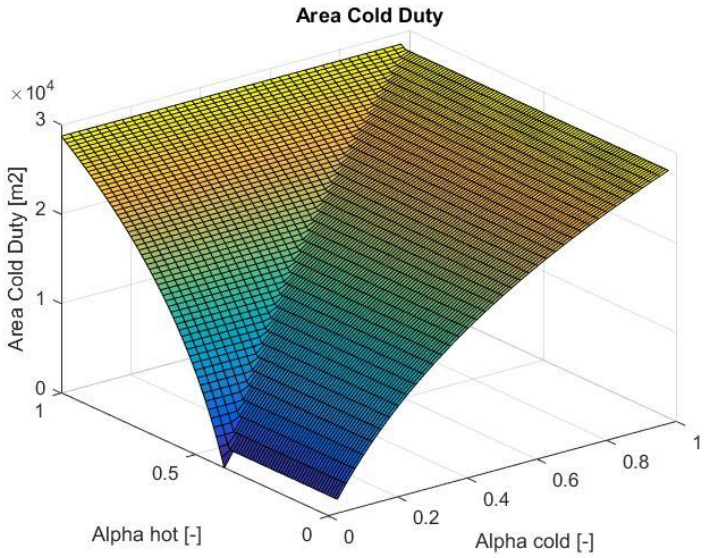


***Figure 31**. (a) Area of hot duty. (b) Area of cold duty.*

Again, the behaviors of these surfaces are essentially the same of the respective graph of costs, but they are also more linear, as the dependence area-split factors is more direct.

Another interesting characteristic to be analyzed is the logarithmic temperature difference ΔT_{ml}, whose graphs obtained on MATLAB® are reported here:

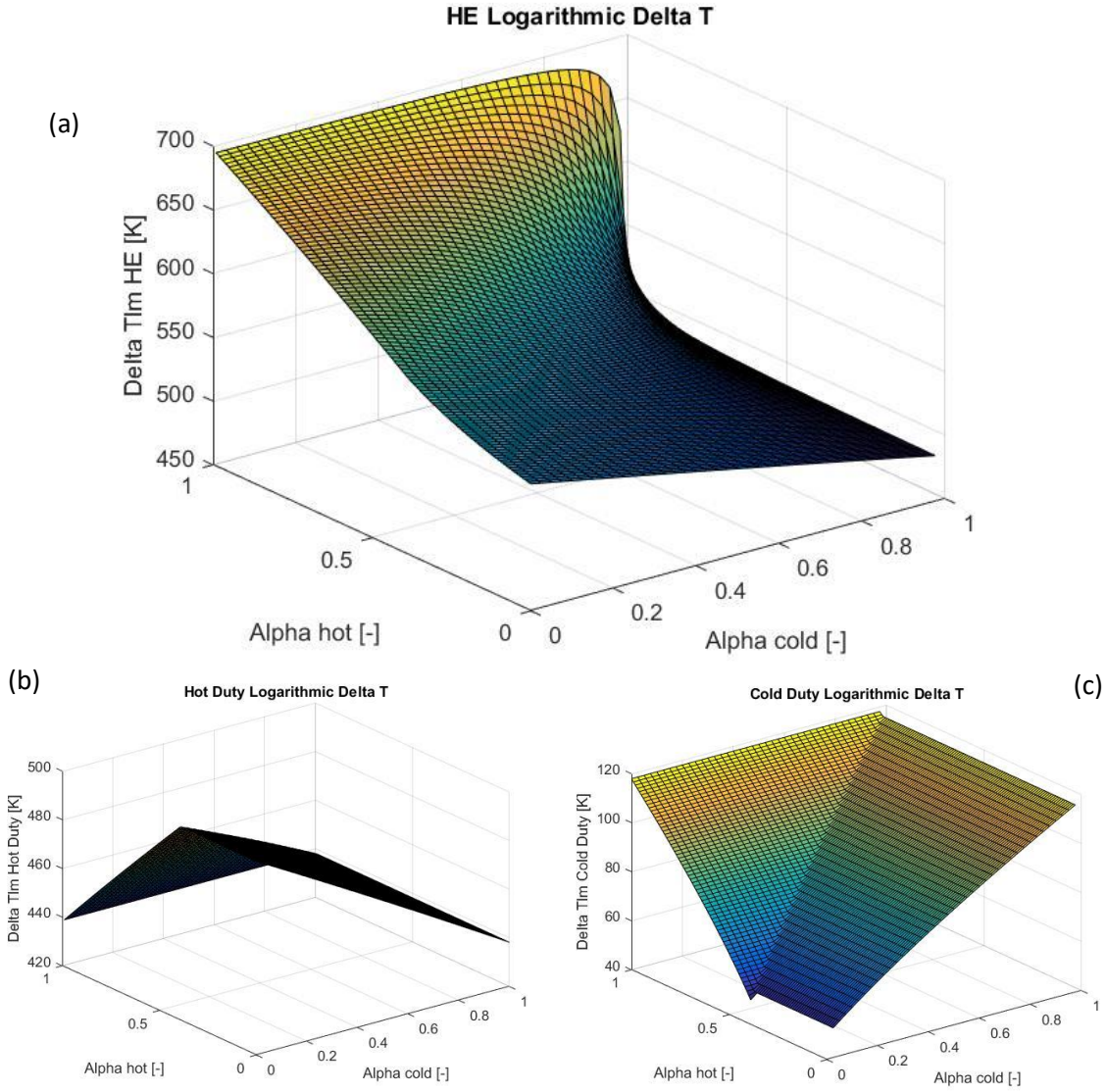


***Figure 32**. (a) Logarithmic temperature difference over heat exchanger unit only. (b) Logarithmic temperature difference over cold duty. (c) Logarithmic temperature difference over cold duty.*

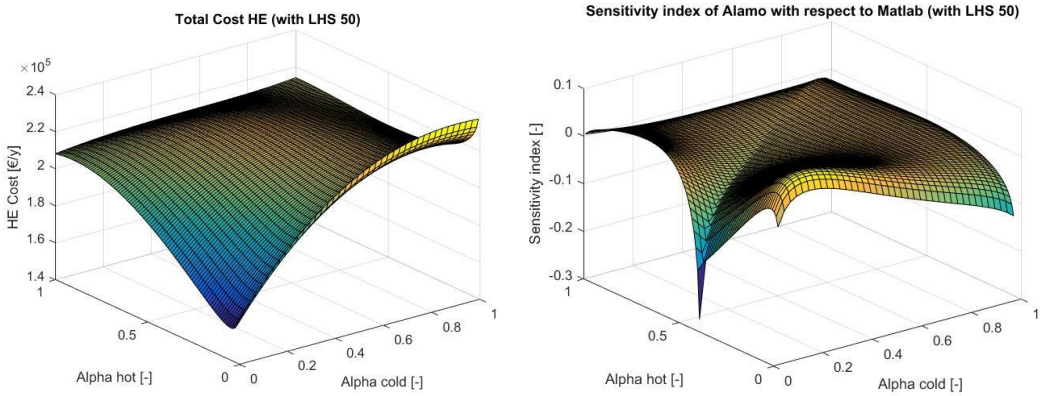
The logarithmic temperature difference relative to both hot and cold duty ($\Delta T^U_{ml_H}$ and $\Delta T^U_{ml_C}$) directly reflects the behavior of the respective costs and area, but with more marked linearity. Moreover, the gap between maximum and minimum is around 60 $[K]$ and 70 $[K]$ and it is less than the one of heat exchanger, as it was presumable, since most of the exchange must occur there in order to have a good optimization, not in the utility units. Finally, the surface of logarithmic temperature difference of the

single heat exchanger (ΔT_{ml}^{HE}) is quite interesting, as it has a different shape from the respective surfaces of cost and area. In fact, it depends on the amount of heat to be transferred, so the larger values are approximatively in the region where heat goes from cold to hot side, more in particular the maximum is along the edge related to $\alpha_{HOT} = 1$, where the whole hot flux is bypassed. Hence, the cold flow should exchange heat with almost nothing, so the temperature difference over the entire unit stays big. Furthermore, there is more linearity toward $\alpha_{COLD} = 0$, while the behavior for $\alpha_{COLD} = 1$ is more quadratic.

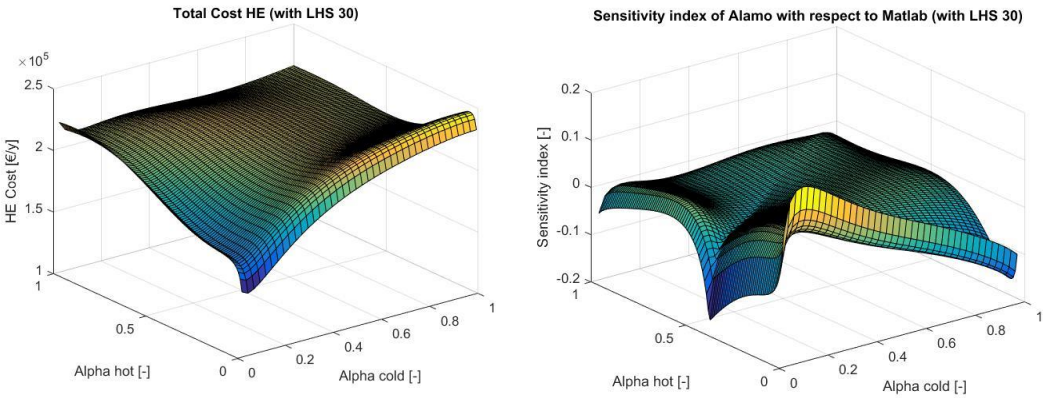
5.1.2 Results of ALAMO

In this section the results obtained by ALAMO will be analyzed. As said before, three different types of sampling techniques have been sequentially exploited and for each one three different sample sizes have been taken, in order to state the dependence of ALAMO from the how training sets are chosen and how many points are necessary to obtain a reliable surrogate model. Only graphs relative to the total cost of the whole system (heat exchanger unit plus hot and cold duties altogether) has been studied and what we obtained is shown throughout this paragraph.

(a)



(b)



(c)

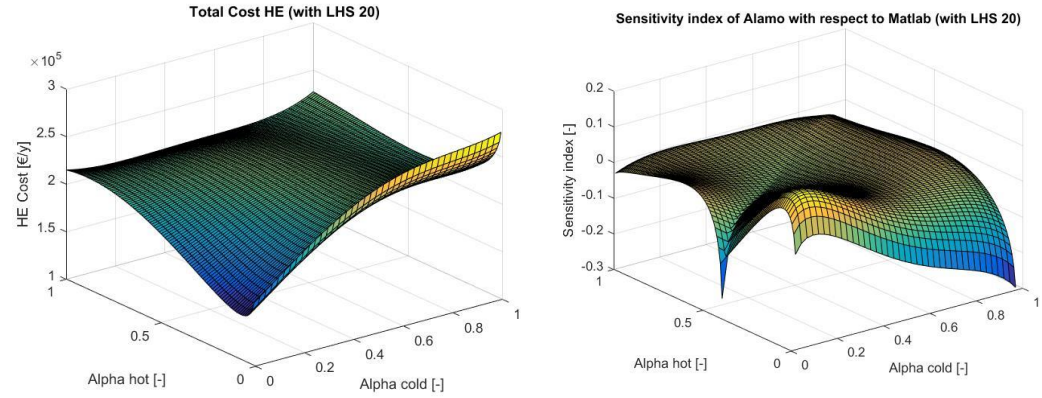


***Figure 33**. Surrogate model of total cost of the entire system (heat exchanger and relative utilities), obtained with Latin Hypercube Sampling and the respective sensitivity index of ALAMO. (a) 50-point sample. (b) 30-point sample. (c) 20-point sample.*

In Figure 34 the surrogate model of total cost relative to the single heat exchanger is reported: the Latin Hypercube Sampling (LHS) was implemented by means of the command $lhsdesign$ already provided by MATLAB®, so that the training sets of points were automatically generated. We decided to work on three different samples: 50 points, 30 points and 20 points.

The original surface has been previously analyzed in figure 25 and consulting it may be very useful while commenting the results of ALAMO. Just watching at the shape, surrogate surface seems to be quite similar to the MATLAB® one for high values of α_{HOT}, but it is a bit different for lower values. In addition, some maximum points have been transformed in minimum points, especially for the 50-point and 20-point samples, hence there is the risk to change the analytical meaning of those surface areas. Instead, considering the respective sensitivity indexes, the biggest sample has the smallest gap between the original and the surrogate results, which is around 10% or less. However, the only exception is related to the most critic region, that presents a cuspid in the MATLAB® surface for $\alpha_{HOT} = 0.4$ and $\alpha_{COLD} = 0$: around this point there is an error of about 30%. In the successive smaller samples, that critical point keeps having the same level of difficulty while being approximated by ALAMO, but the error over the entire surface is globally increased. In effect, this observation matches with the shapes of the respective surrogate models, which are more different from the original one as they appear more planar, thus tending to equalize maximum to minimum points. It is very likely that the decreasing accuracy depends a lot on how many training points are available for the regression: so, the less data are selected at the beginning, the worse the surrogate will be.

Next graphs from ALAMO to be analyzed are relative to the second sampling technique mentioned above: Cluster Sampling (CS). In a similar way of before, different sample sizes were sequentially adopted: 7 clusters of 5 points each one, 6 clusters of 4 points each one and 6 clusters of 3 points each one. Therefore, the resulting surfaces are shown in Figure 35.

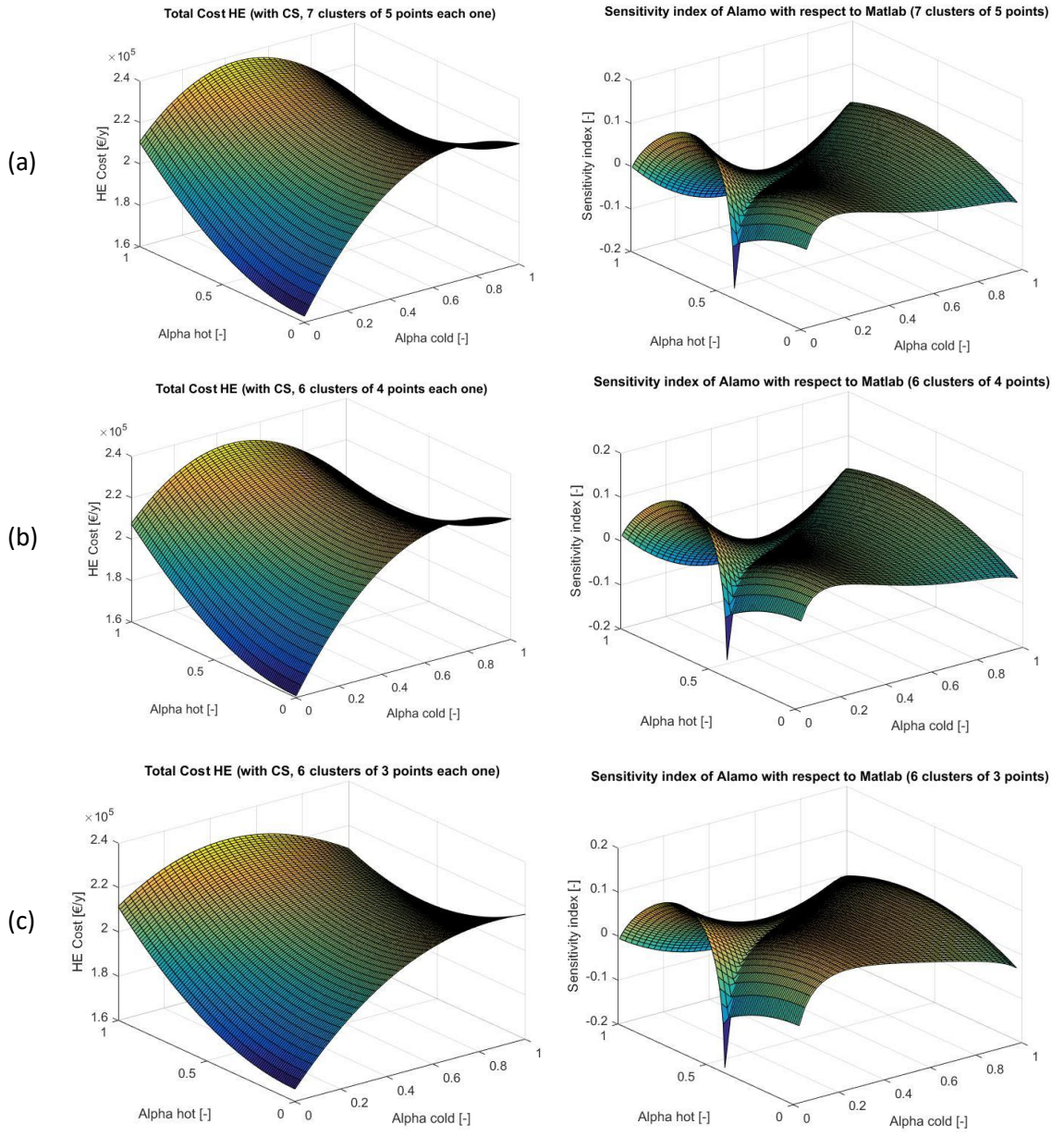


Figure 34*. Surrogate model of total cost of the entire system (heat exchanger and relative utilities), obtained with Cluster Sampling and the respective sensitivity index of ALAMO. (a) Sample of 7 clusters of 5 points each one. (b) Sample of 6 clusters of 4 points each one. (c) Sample of 6 clusters of 3 points each one.*

It is immediately observable that the surrogate surface does not change a lot its shape and likely the accuracy even reducing the total amount of training points, so the dependence from the quality of prior sampling is less than the case of LHS. With all three sampling, the shapes are tendentially more planar than the original one, thus some points may lose or slightly modify their importance during a hypothetical successive analysis. For example, some local maximum points have become local minimum points. Moreover, the curves are quite similar one to each other, thus expressing that this

sampling technique is not too much influenced by the change of the size of training sets and it may be more robust than others. Again, a similar behavior appears from sensitivity indexes, as it stays practically equal for all three sample sizes: the surface region around $\alpha_{HOT} = 0.4$ and $\alpha_{COLD} = 0$ still presents a critic issue for the regression in ALAMO, but in the rest of the surface the error is globally about $\pm 10\%$.

The last implemented sampling technique is Systematic Sampling (SS) (shown in Figure 36), which was sequentially studied for three different sample sizes: a 6×5 grid (30 points), a 6×4 grid (24 points) and a 4×4 grid (16 points). It just consists of a grid of sample points distributed over the entire surface, without guiding choice in any direction, as it was made for the Cluster Sampling. In figure 32 there are the resulting graphs and the relative sensitivity indexes. Unfortunately, Systematic Sampling has given surrogate models quite different from the original surface, even if the resulting sensitivity index showed a global error approximatively in the range $[-10\%, +10\%]$, except for the usual region around $\alpha_{HOT} = 0.4$ and $\alpha_{COLD} = 0$, where it goes up to -30%. Effectively, the strangest feature of these three curves is that they vary along α_{COLD} only and they are constant along α_{HOT}. It is not necessary to comment every single sample size as they are almost identical, meaning that this sampling technique is not influenced too much by the amount of training points, so it is a robust method but also extremely few flexible. It probably needs a more detailed grid when there are complex surfaces to be approximated, as they may contain a lot of local maxima or minima which are difficult to be accounted for by a very fix system like this one.

(a)

(b)

(c)

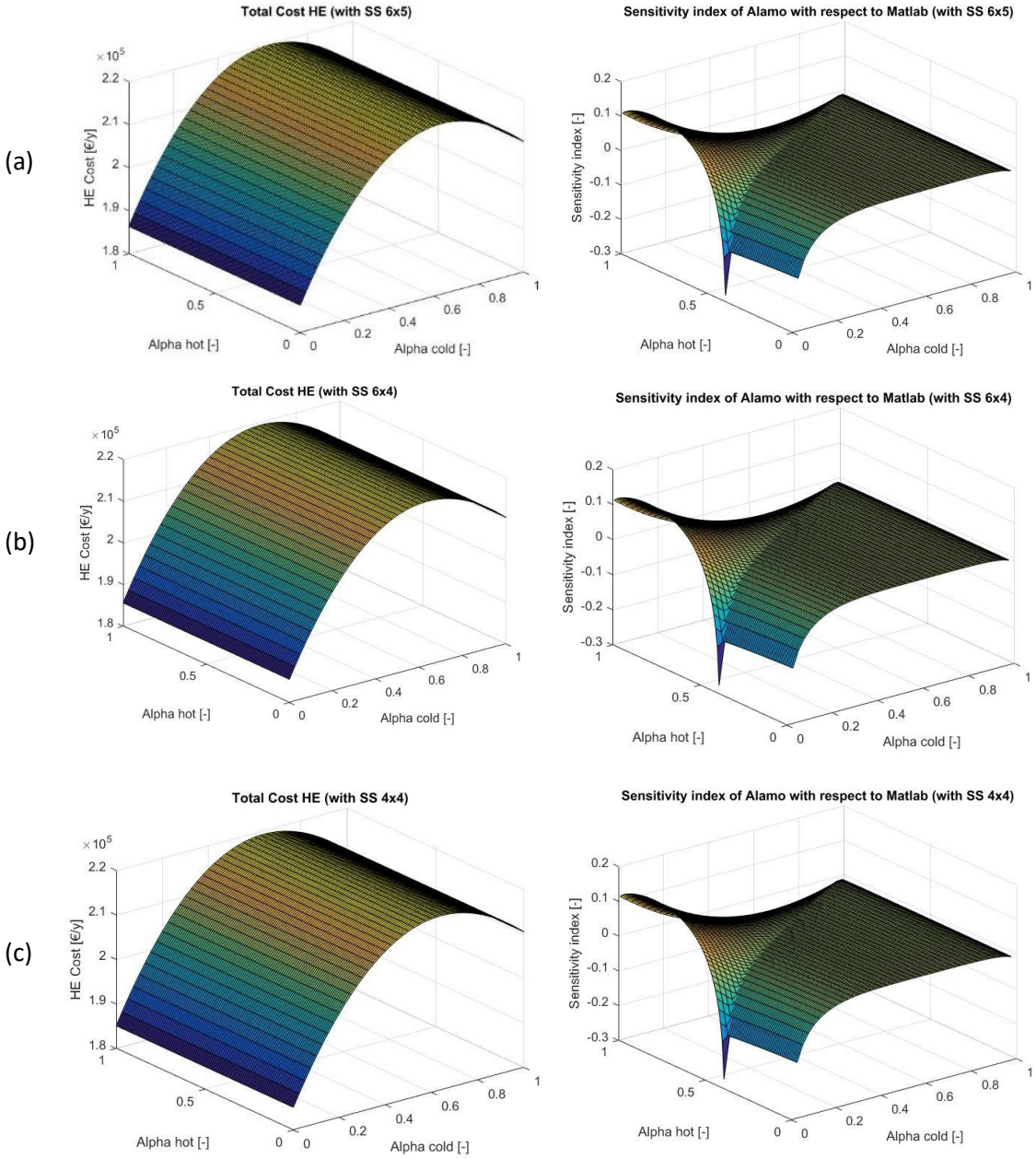


***Figure 35**. Surrogate model of total cost of the entire system (heat exchanger and relative utilities), obtained with Systematic Sampling and the respective sensitivity index of ALAMO. (a) Grid of 6 × 5 points. (b) Grid of 6 × 4 points. (c) Grid of 4 × 4 points.*

5.2 Two counter-current heat exchangers

The code used to simulate this case is not so different from the previous one, but it is just longer as the IF-cycles must be replicated twice instead of once. Of course, some additional data are required, since there are more flows and more utilities. Furthermore, the split factors are now four, hence four independent variables which give back four-dimension results: two on the hot side (α_{HOT}^{1} and α_{HOT}^{2}) and two on the cold side (α_{COLD}^{1} and α_{COLD}^{2}). In addition, the range of each split factor α_i is $[0,1]$ and it has been divided into 50 points for each one, because the amount of data to be elaborated by the computer starts to be quite abundant, so it has been necessary to reduce it a bit in order to have good results in reasonable computational time. The total cost of the entire system has been analyzed, thus CAPEX and OPEX for all the units together, but it is not so simple to represent 4D results. Therefore, the independent variables have been made vary twice per time, leaving the remaining two as constant and using three different values of these two each time in order to appreciate any variation, like three different "slices" of the 4×4 matrix. The graphs are reported in Figure 37, Figure 38, Figure 39 and Figure 40.

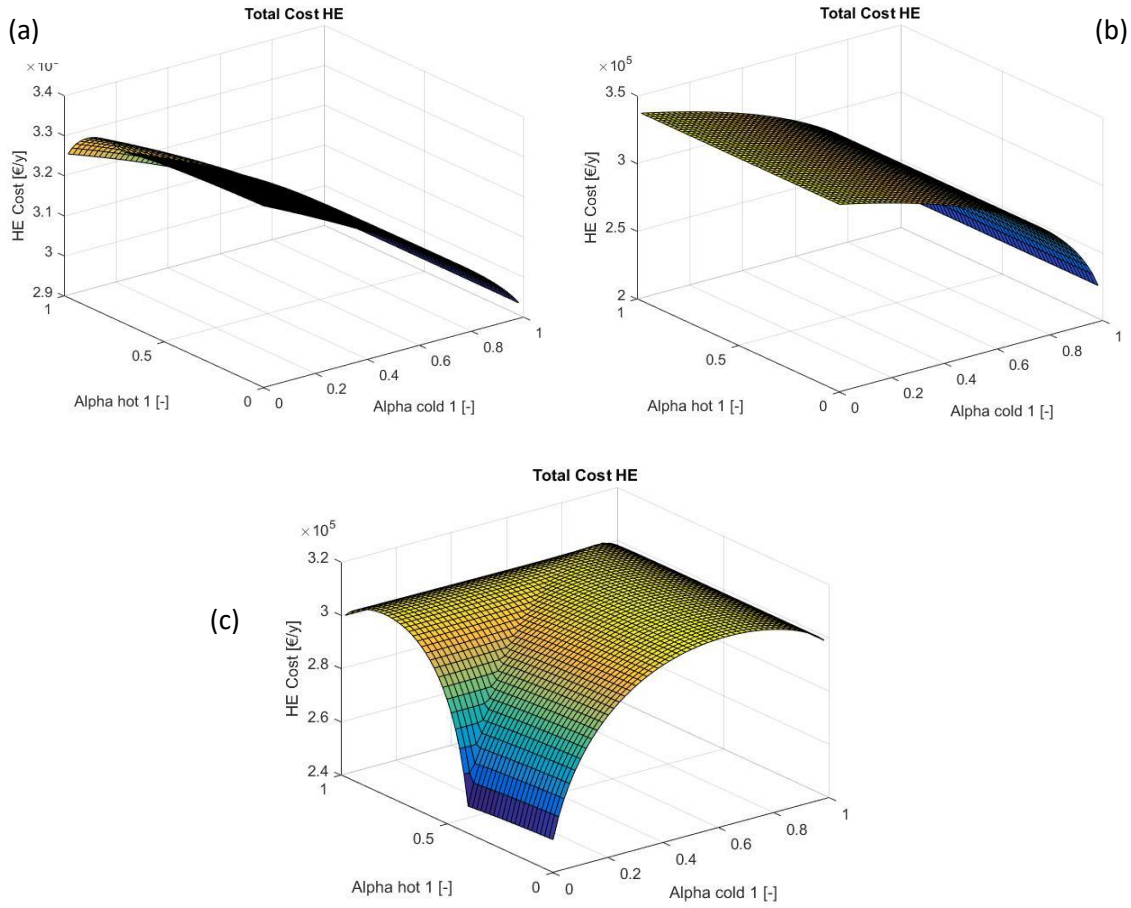


***Figure 36**. Total cost of two counter-current heat exchangers network by changing α_{HOT}^{1} and α_{COLD}^{1}. (a) $\alpha_{HOT}^{2} = \alpha_{COLD}^{2} = 0$. (b) $\alpha_{HOT}^{2} = \alpha_{COLD}^{2} = 0.50$. (c) $\alpha_{HOT}^{2} = \alpha_{COLD}^{2} = 1$.*

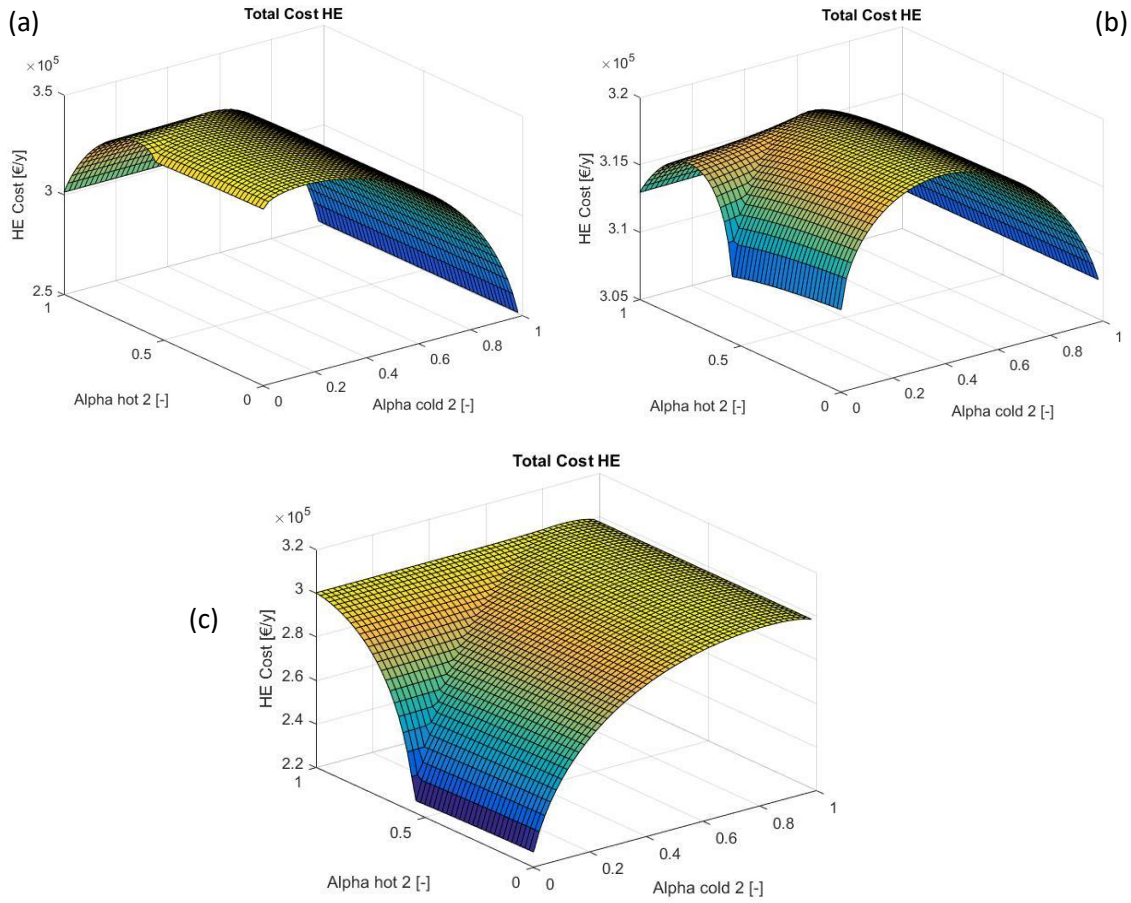


***Figure 37**. Total cost of two counter-current heat exchangers network by changing α_{HOT}^{2} and α_{COLD}^{2}. (a) $\alpha_{HOT}^{1} = \alpha_{COLD}^{1} = 0$. (b) $\alpha_{HOT}^{1} = \alpha_{COLD}^{1} = 0.50$. (c) $\alpha_{HOT}^{1} = \alpha_{COLD}^{1} = 1$.*

When $\alpha_{\mathrm{HOT}}^{1}$ varies with respect to $\alpha_{\mathrm{COLD}}^{1}$ (Figure 37) and $\alpha_{\mathrm{HOT}}^{2}$ with respect to $\alpha_{\mathrm{COLD}}^{2}$ (Figure 38), the behavior of the total cost is quite similar to the case of single heat exchanger: this happens likely because these two combinations are essentially equal to the base case, hence it is like there two separated heat exchangers which have exactly the same functioning. It is different for middle combinations as their behaviors change. In fact, for $\alpha_{\mathrm{HOT}}^{1}$ changing with $\alpha_{\mathrm{COLD}}^{2}$ (Figure 39), the surface is equal to the previous one, but symmetrically reflected with respect to the $\alpha_{\mathrm{HOT}}^{1}$-axis. Instead, for the last combination $\alpha_{\mathrm{HOT}}^{2}$ changing with $\alpha_{\mathrm{COLD}}^{1}$ (Figure 40), the shapes are totally different: indeed, when $\alpha_{\mathrm{HOT}}^{1} = \alpha_{\mathrm{COLD}}^{2} \cong 1$, the total cost is constant as $\alpha_{\mathrm{HOT}}^{2}$ varies, but it also linearly decreases with increasing $\alpha_{\mathrm{COLD}}^{1}$, except for a local minimum for $\alpha_{\mathrm{COLD}}^{1} \cong 1$.

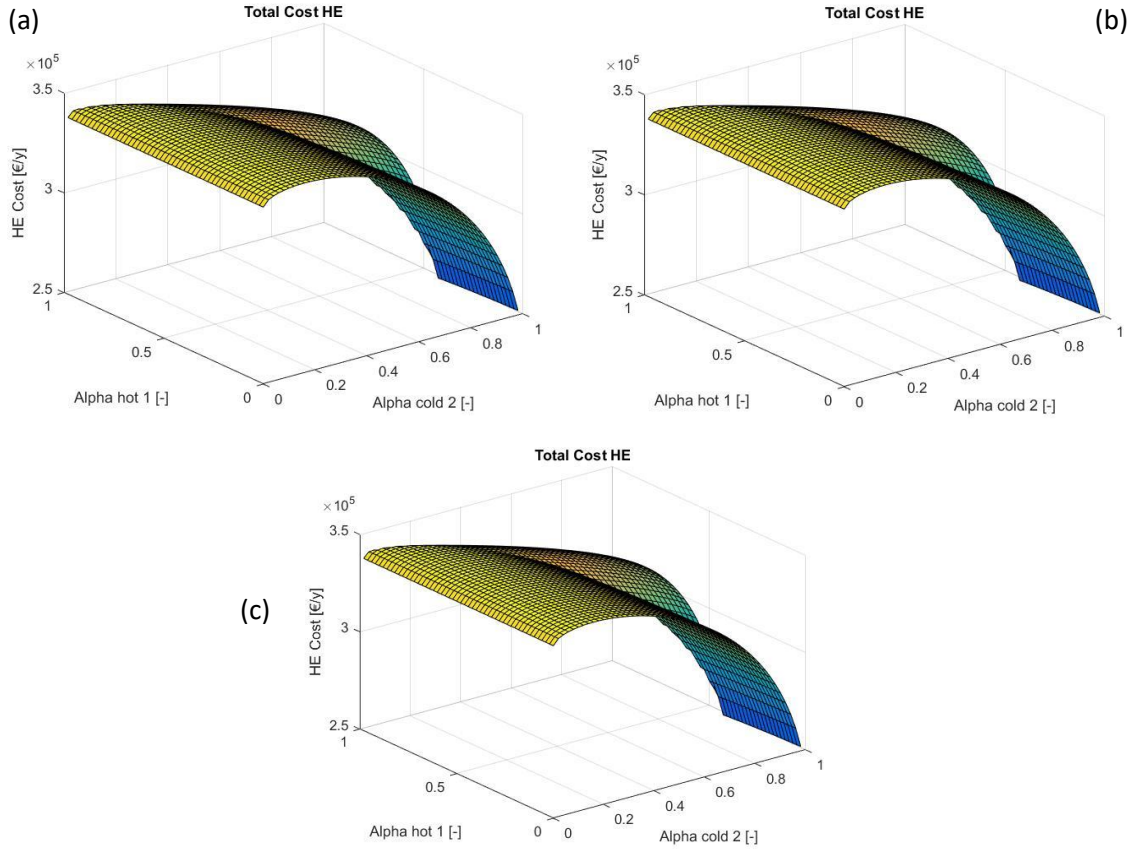


***Figure 38**. Total cost of two counter-current heat exchangers network by changing α_{HOT}^{1} and α_{COLD}^{2}. (a) $\alpha_{HOT}^{2} = \alpha_{COLD}^{1} = 0$. (b) $\alpha_{HOT}^{2} = \alpha_{COLD}^{1} = 0.50$. (c) $\alpha_{HOT}^{2} = \alpha_{COLD}^{1} = 1$.*

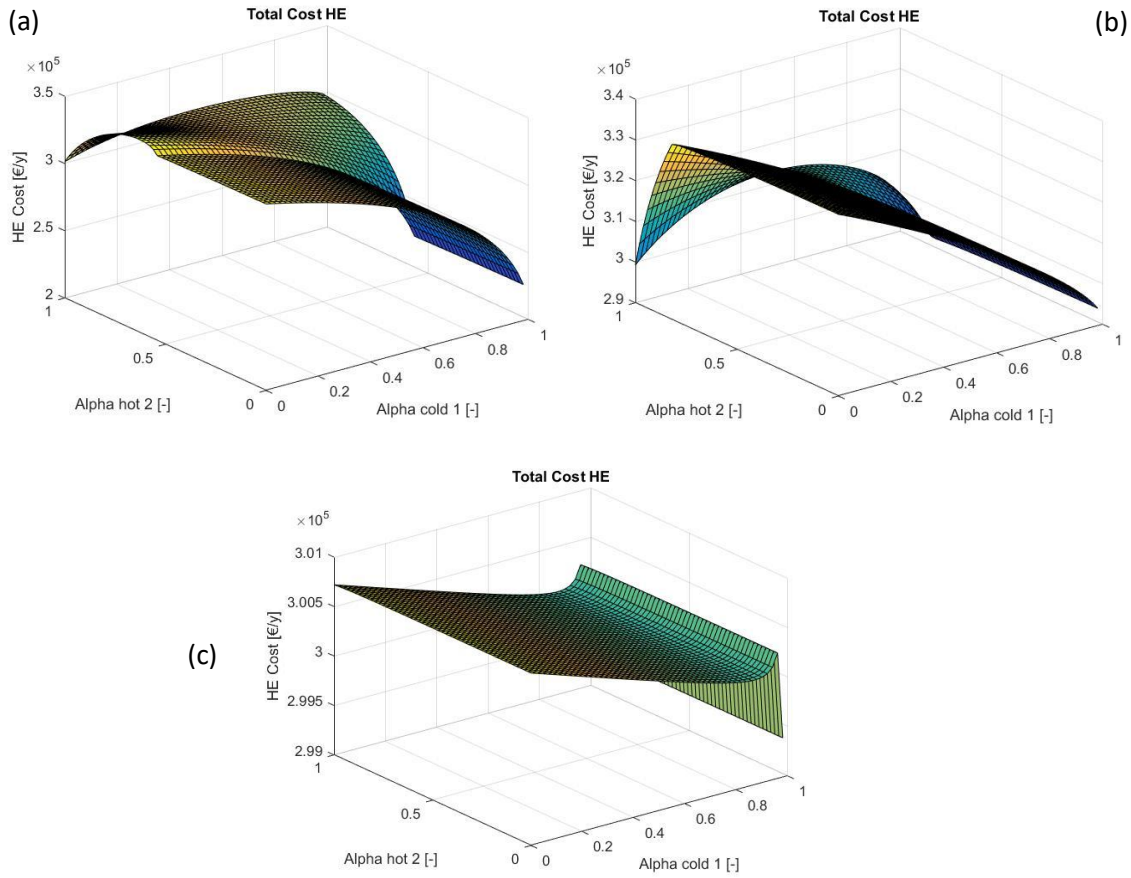


***Figure 39**. Total cost of two counter-current heat exchangers network by changing α^2_{HOT} and α^1_{COLD}. (a) $\alpha^1_{HOT} = \alpha^2_{COLD} = 0$. (b) $\alpha^1_{HOT} = \alpha^2_{COLD} = 0.50$. (c) $\alpha^1_{HOT} = \alpha^2_{COLD} = 1$.*

5.3 Three counter-current heat exchangers

Again, there are two main differences in this third MATLAB® code, that is the IF-cycle is used three times, one for each heat exchanger unit, and the split factors become six. Therefore, there are six independent variables: two on the first hot flow (α_{HOT}^{1} and α_{HOT}^{2}), two on the first cold flow (α_{COLD}^{1} and α_{COLD}^{3}), one on the second hot flow (α_{HOT}^{3}) and one on the second cold flow (α_{COLD}^{2}). So, In the end all results will be in six dimensions, thus more complicated to be analyzed with respect to the two-heat exchanger case.

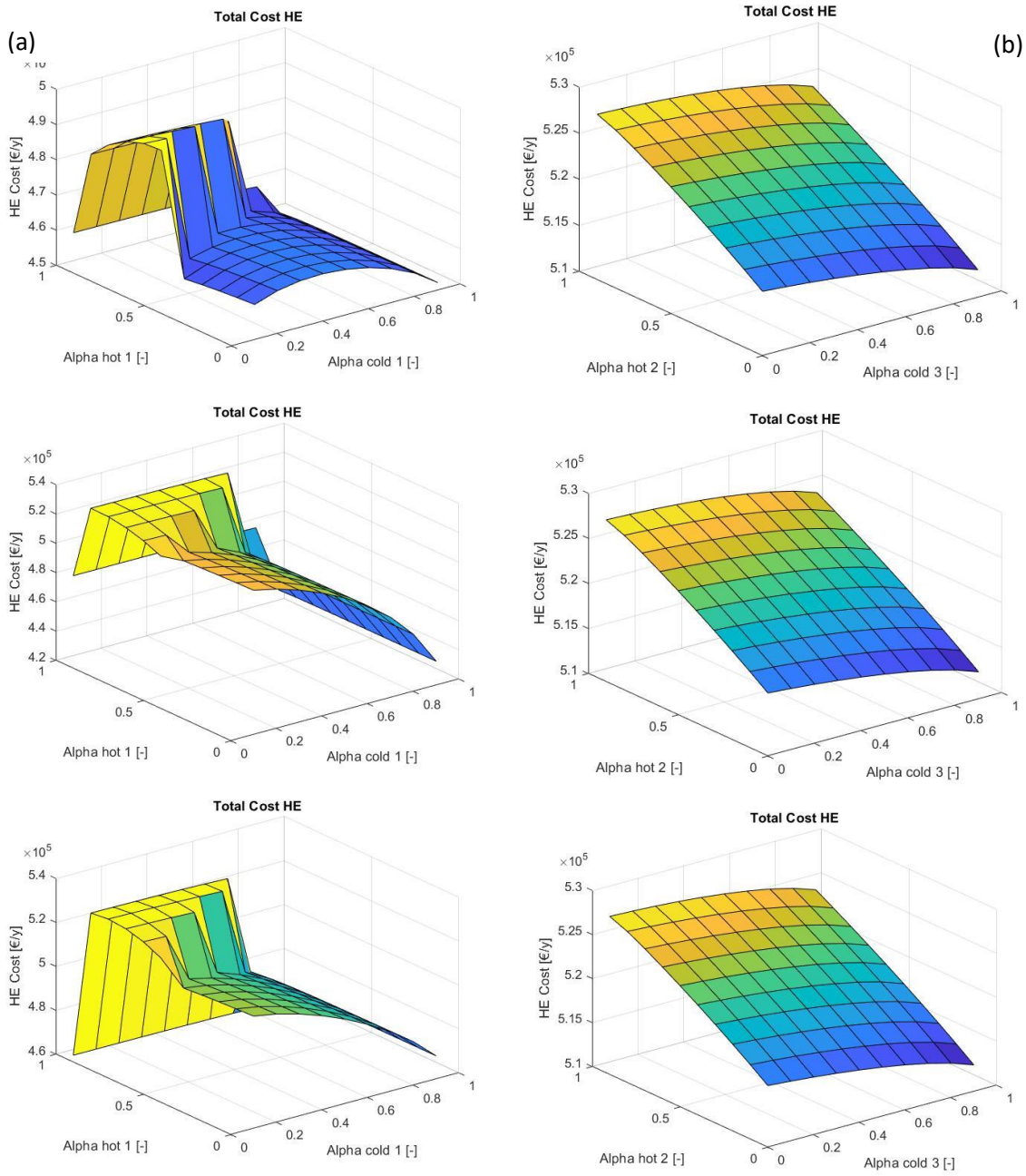


***Figure 40**. Total cost of three counter-current heat exchangers network. (a) By varying α_{HOT}^{1} and α_{COLD}^{1}. (b) By varying α_{HOT}^{2} and α_{COLD}^{3}.*

For the sake of simplicity, since split factors α_i are six, they have been divided into couples composed of one α_i on the hot side and the other one on the cold side, so that one varies along with the other while keeping the remaining α_i constant. Therefore, we obtained six different couples and for each one we reported three graphs: one for constant α_i near to 0, one for constant α_i equal to 0.5 and one for constant α_i near to 1.

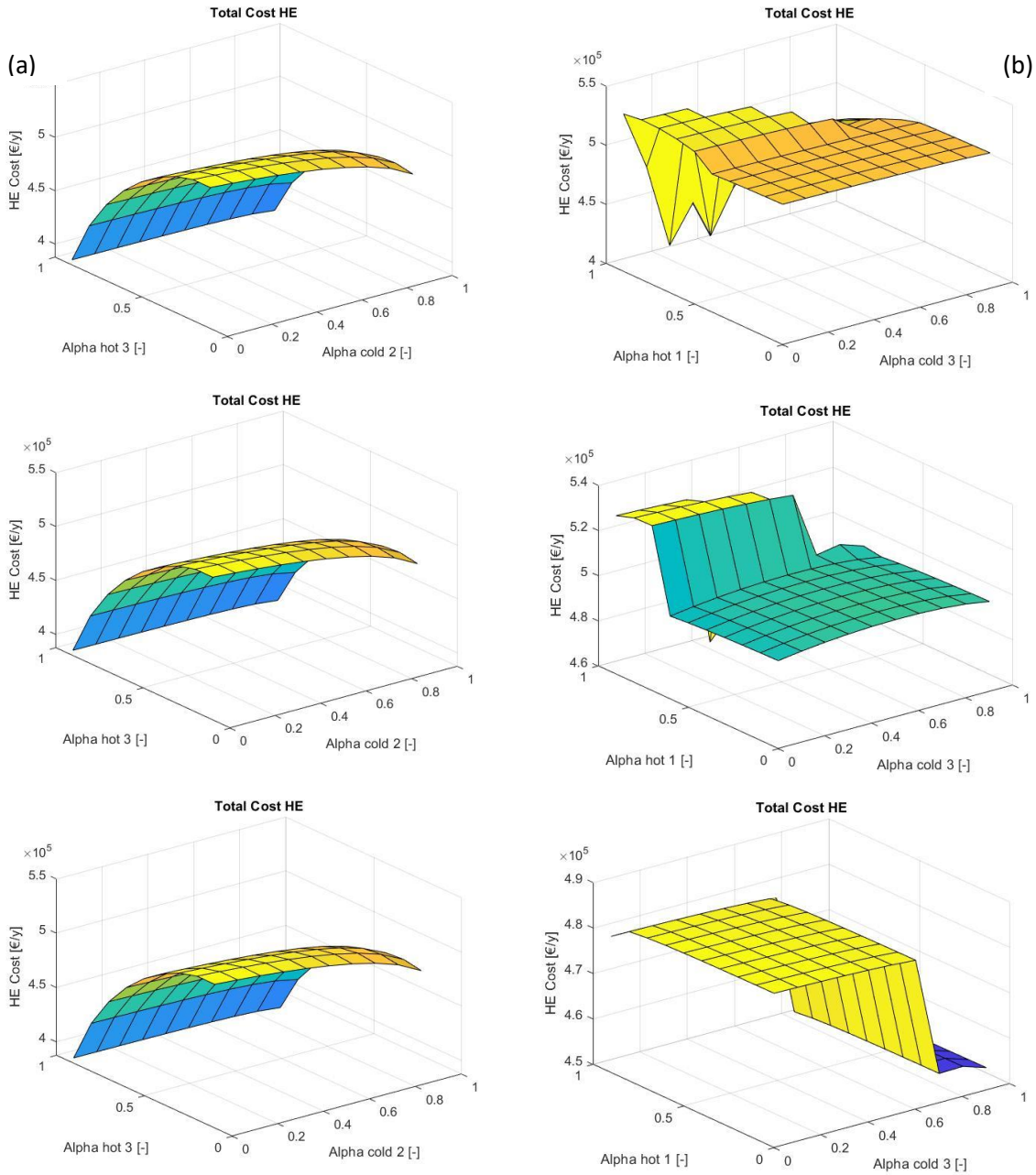


Figure 41. *Total cost of three counter-current heat exchangers network. (a) By varying α_{HOT}^{3} and α_{COLD}^{2}. (b) By varying α_{HOT}^{1} and α_{COLD}^{3}.*

We decided to avoid considering $\alpha_i = 0$ and $\alpha_i = 1$ because they are a sort of boundary conditions, so there could be limit behaviors like asymptotes or complex values. This strategy allows us to have an idea of how the total cost evolves by changing just one couple of split factors, even if it is difficult to have a global view over the system. One thing that may be easy to recognize is the dependence between one α_i and another, which can be linear (like α^2_{HOT} vs. α^3_{COLD}) or more complex (like α^1_{HOT} vs. α^1_{COLD}).

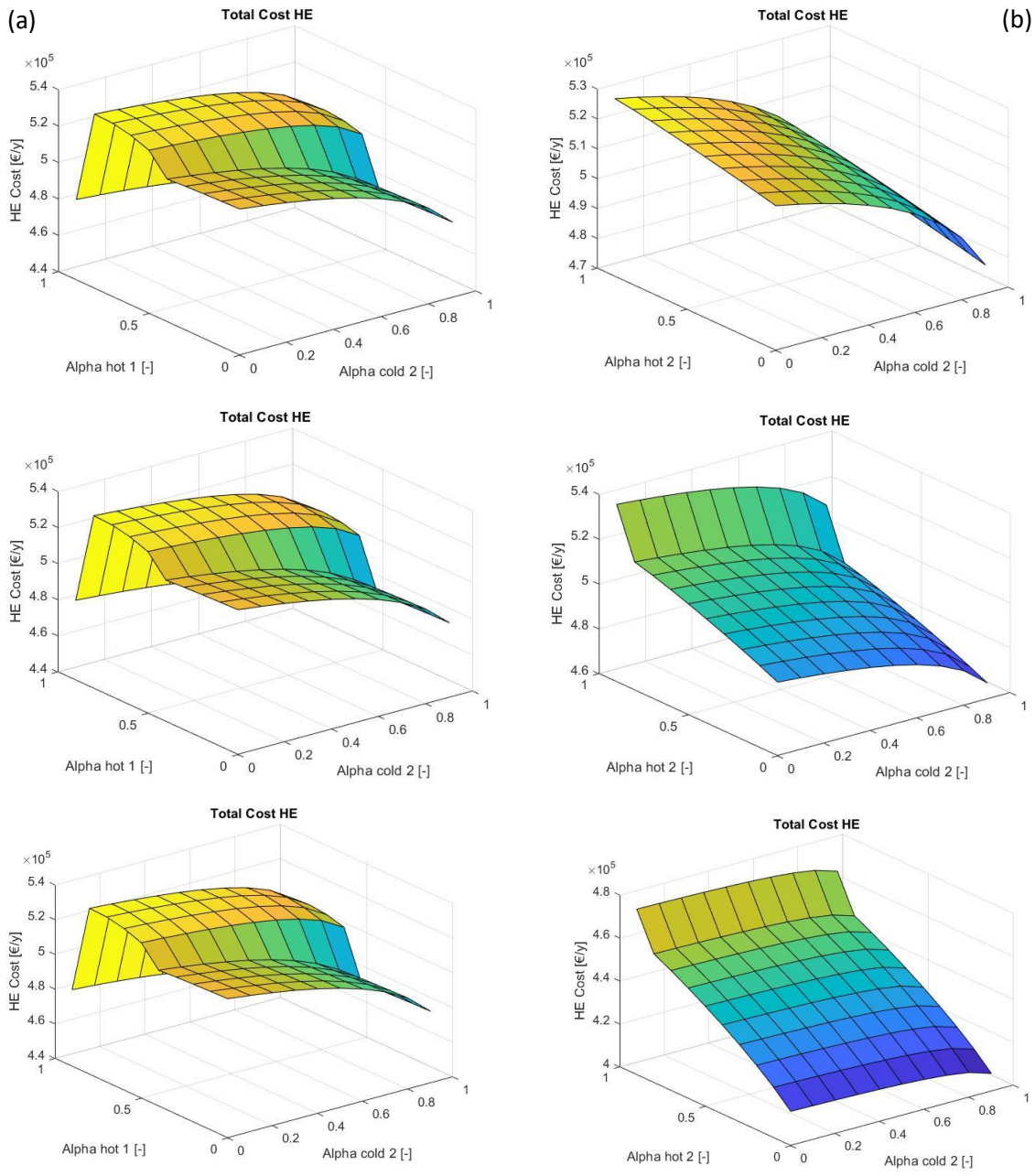


Figure 42. *Total cost of three counter-current heat exchangers network. Left: by varying α^1_{HOT} and α^2_{COLD}. Right: by varying α^2_{HOT} and α^2_{COLD}.*

5.4 Four counter-current heat exchangers

Finally, the results relative to the four-heat exchanger system are reported in this section. Basically, it represents the Grossmann's superstructure introduced at the beginning, which has been slightly modified in order to be compatible with the type of computations in MATLAB® environment. As it happened for the second and third level of simulation, the IF-cycle is repeated for each heat exchanger, thus four times. So, the split factors used in the code are eight: two on the first hot flow (α_{HOT}^{1} and α_{HOT}^{2}), two on the first cold flow (α_{COLD}^{1} and α_{COLD}^{3}), two on the second hot flow (α_{HOT}^{3} and α_{HOT}^{4}) and two on the first cold flow (α_{COLD}^{2} and α_{COLD}^{4}).

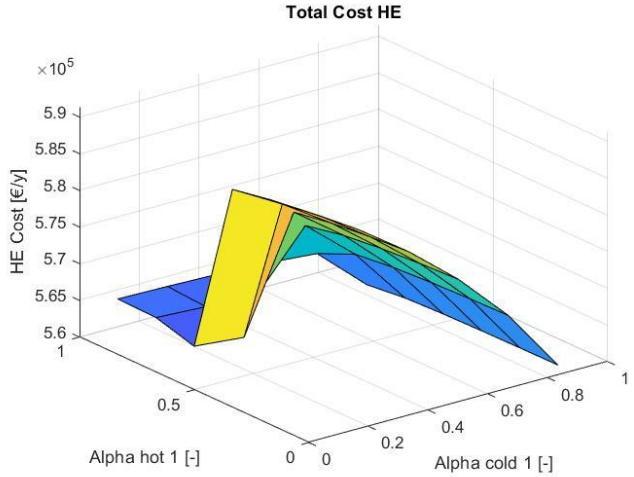


***Figure 43**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{1} and α_{COLD}^{1}.*

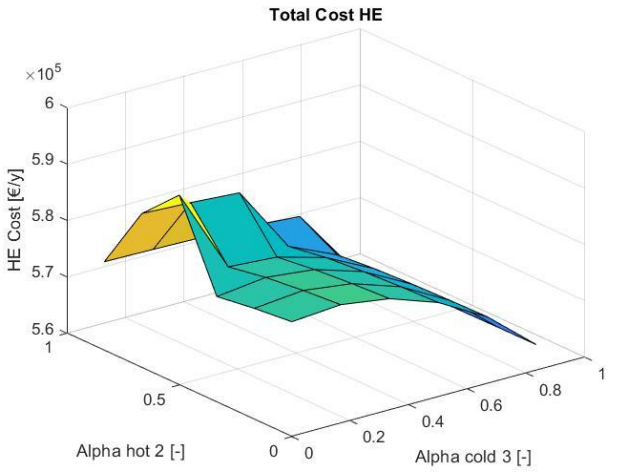


***Figure 44**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{2} and α_{COLD}^{3}.*

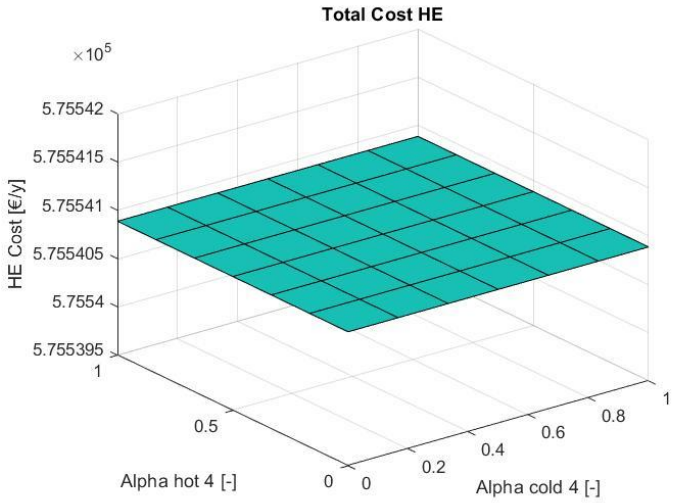


***Figure 46**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{4} and α_{COLD}^{4}.*

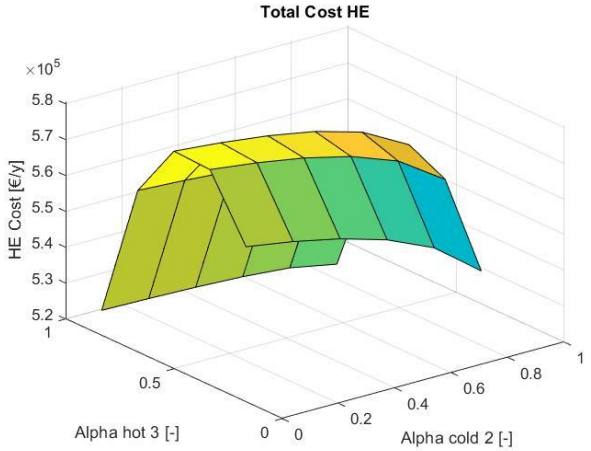


***Figure 45**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{3} and α_{COLD}^{2}.*

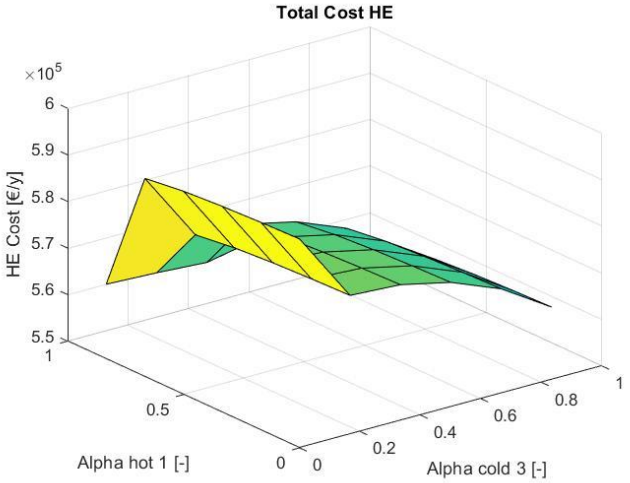


***Figure 47**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{1} and α_{COLD}^{3}.*

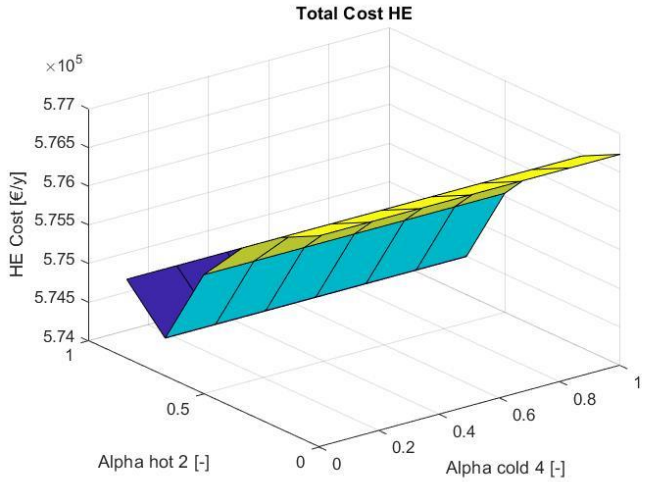


***Figure 49**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{2} and α_{COLD}^{4}.*

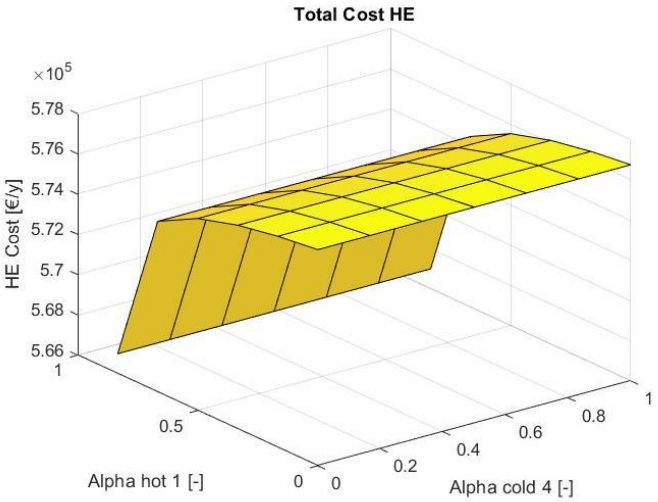


***Figure 48**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{1} and α_{COLD}^{4}.*

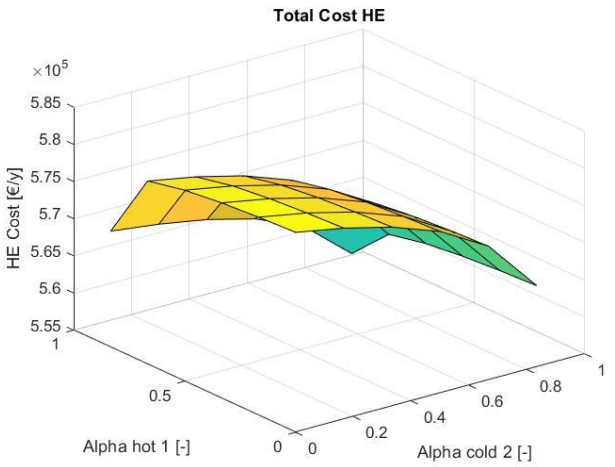


***Figure 50**. Total cost of four counter-current heat exchangers network by varying α_{HOT}^{1} and α_{COLD}^{2}.*

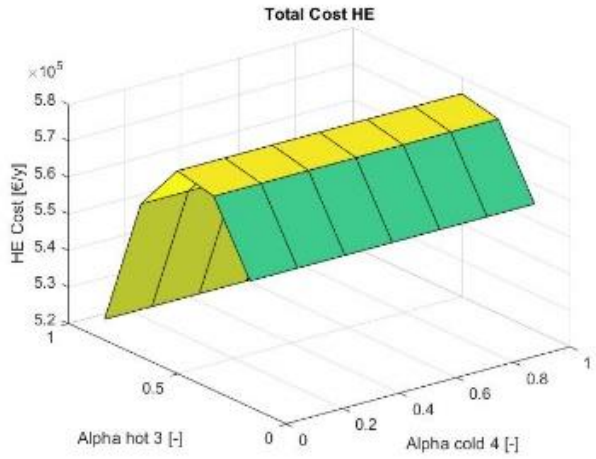


Figure 51. *Total cost of four counter-current heat exchangers network by varying* α_{HOT}^{3} *and* α_{COLD}^{4}.

By observing those images, accuracy and smoothness are strongly reduced as the discretization of vectors α_i is smaller. In fact, each α_i has been divided into 6 points instead of 100 or 50 of previous simulations and the reason is quite simple: the amount of computational efforts for a normal notebook to be elaborated is huge. Anyway, what is immediately clear is the reciprocal dependence of one split factor from the other inside each couple. For example, α_{HOT}^{4} and α_{COLD}^{4} seem to be not directly connected; there can a complex behavior with maxima points, like α_{HOT}^{1} vs. α_{COLD}^{1} and α_{HOT}^{2} vs. α_{COLD}^{3} cases; otherwise, some curves show a quadratic dependence for from one split factor and total cost while the other α_i has no influence on total cost, like for α_{HOT}^{3} and α_{COLD}^{4}. The highest cost found in these graphs is in Figure 48, where it is slightly larger than 5.8×10^5 €$/y$; instead, the lowest cost is 5.2×10^5 €$/y$ in Figure 52, so this is the range in which appropriate managements on bypass valves should be done in order to approach the minimum cost as much as possible.

6 CONCLUSIONS

As we said at the beginning of this thesis, the first target of this work is to find a way to optimize the Grossmann's superstructure through less computational efforts than actual techniques. Consequently, the second task is to test an experimental strategy that generates surrogate models, whose main function is to approximate the obtained simulation data as best as possible in order to mathematically describe the reality, thus allowing future improvements and reducing posterior research costs.

Regarding the optimization of the heat exchanger network, the results are different for each of the four cases. In the single heat exchanger, the graph total cost reveals that the maximum points are along the lines relative to α_{HOT} or α_{COLD} and their intersection, while the minimum points stay on the edge of the surface where $\alpha_{COLD} = 0$, in a range of α_{HOT} going from 0 to about 0.4. In addition, it is possible to note that total expense depends more from CAPEX than OPEX, as the duties needed to reach the required specifications are way less than the successive cases.

For the two-heat exchanger configuration, the behavior of surfaces is globally similar to the first case, while the amount of necessary money is obviously augmented, as the number of heat exchanger units and duties are more, even if it is less than the double.

Instead, the results relative to three- and four-heat exchanger cases are extremely complex to be analyzed, due to their nature. In fact, the major difficulty is to convert 6D and 8D numbers into 2D, which is a mandatory action for their visual representation as curves in order to be studied later. Consequently, it is hard to identify the main features of the obtained graphs and to describe their behavior. Anyway, in the configuration with three heat exchangers it is possible to define the dependences among all split factors and what we can surely say is that the total cost is approximatively comprised between 4.1×10^5 and 5.5×10^5 €$/y$.

In the end, regarding the final simulated system, that is the one which approximates Grossmann's superstructure, it is possible to state that the range of total cost is more or less between 5.2×10^5 and 5.8×10^5 €$/y$. Of course, it is the most expensive configuration as the number of heat exchangers is the biggest one, but the behavior of global expense with respect to all varying split factors is quite difficult to be defined. Hence, it is hard to have a global overview on the results in order to clarify and address all the optimization, as they are not easy to be interpreted. At least, we can say that the obtained costs are likely and plausible, so this preliminary work may be the first step of successive studies, which will probably insert more realistic details and parameters, fix some unavoidable inaccuracies due to the assumptions and improve this methodology by starting from the strategy used here.

One relevant issue was the computational power of the available instruments, which must be adequate to the needed efforts, as the dimension of matrixes in the last code has been in the order of billions and billions of single data and elementary mathematical operations to be elaborated. Therefore, future researches should be made by means of more powerful devices, which are largely available nowadays in universities and companies.

Regarding the surrogate modeling work, the results we attained seem to be newsworthy, as the error has been found to be globally in the order of about 10%, except for the most critical regions of the original function, where it may reach almost 30%. In these areas of curves, there may be asymptotes or cuspids, so that it is harder for ALAMO to make the regression and approximate adequately the behavior of original data. However, it is important to remark that not all the data are good-looking: in fact, ALAMO strictly depends on efficiency of the prior sampling technique used to obtain training data because of the intrinsic structure of its algorithms. For example, the Systematic Sampling resulted to be inappropriate, due to its strong rigidity, while Latin Hypercube Sampling has shown good qualities as it is random, more flexible and more suitable for complex shapes. Besides, the Cluster Sampling is quite good thanks to its adaptability to many objective functions, but it also depends too much on how the modeler chooses the clusters of training points, hence it is too subject to variations and it exploits less the power of computation of computers. Therefore, the Latin Hypercube Sampling has been found to be the most satisfying one that has been tried in this work. ALAMO has a good potential since it is based on simple principles (see paragraph 2.10 for every explanation), thus this feature makes this software easy to run even on a personal computer and without problems like overall huge computational efforts. It is also suitable to many problems and many different behaviors, due to its step-wise adaptability. Actually, the results achieved in this thesis work are promising and cheering, even if they show relevant inaccuracies and they still are not good enough for any real optimization. In effect, the whole study faced in this work has analyzed the potentialities of this innovative technique, which is still at the experimental step, thus it is just the starting point for more detailed researches in the future.

There are a lot of things that should be fixed, such as the sampling techniques. As said above, ALAMO may be an efficient “machine” for surrogate models, but it is extremely dependent from those techniques: that’s why it should be supported by algorithms or mathematical methods that aim to more intelligent samplings of training data. One way could be to mix flexibility and randomness typical of Latin Hypercube Sampling with a sampling more directed to critical regions, like Cluster Sampling, with a sort of informatic “brain” which decides where to take new points to be used for the next surrogate. In this way, computational efforts are more smartly distributed, that is samplings occur more intensively where they are needed instead of where the objective function is easy to be approximated. By doing this, ALAMO can operate on better quality training sets, thus improving its

performance and generating a good surrogate model, which may be very useful for successive studies such as the optimization itself.

In effect, improving the prior simulation mechanism of heat exchanger network first and the sampling method for ALAMO regression afterward may be highly interesting. The best thing we can do is not to work at those two issues separately, but to consider both, as one influences the other and vice versa, so that they reciprocally support themselves throughout all optimization steps. Hence, it is worth to investigate more deeply this area of optimization of process units, since the results may be extremely advantageous in economics, time and reliability.

REFERENCES

1. *Oil&Gas Journal, vol. 117, 4a, 8th April 2019*.
2. Alison Cozad et al. (2014). A combined first-principles and data-driven approach to model building. *Computers and Chemical Engineering, 73*, 116-127.
3. Alison Cozad et al. (2014). Automated Learning of Algebraic Models for Optimization. *AIChE Journal, 60*, 2211-2227.
4. Atharv Bhosekar et al. (2017). Advances in surrogate based modeling, feasibility analysis and optimization: a review. *Computers and Chemical Engineering, 108*, 250-267.
5. Boilers, B. (2015). *Boiler Accessories – Heat Exchangers*. Tratto da www.bryanboilers.com: https://bryanboilers.com/boiler-accessories/heat-exchangers/
6. Boukouvala F., F. C. (2016). ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. *Optimization Letters - Springer*.
7. Powell, D. M. (1994). A direct search optimization method that models theobjective and constraint functions by linear interpolation. *Springer Netherlands*, 51-67.
8. Banca d'Italia, *Tassi di cambio - Latest Rates*. Tratto da Banca d'Italia - Eurosistema: https://tassidicambio.bancaditalia.it/latestRates, updated at 29th March 2019
9. Grossmann I. E., Yee, T. (1990). Simultaneous optimization models for heat integration - II. Heat exchanger network synthesis. *Computers Chemical Engineering, 14*(10), 1165-1184.
10. Guthrie, K. M. (1969). *Capital Cost Estimating.* Chemical Engineering.
11. Guthrie, K. M. (1974). *Capitol Cost Estimating, Evaluation and Control.* Craftsman Book Co. of America.
12. Huyer W. et al. (2008). SNOBFIT - Stable noisy optimization by branch and fit. *ACM Trans. Math. Softw., 35*, 1-25.
13. Incropera, D. B. (2007). Fundamentals of Heat and Mass Transfer. 670-672.
14. Industriais, S. (2019). www.solucoesindustriais.com.br: https://www.solucoesindustriais.com.br/
15. Industry, A. (2019). *Storage tank - U tube heat exchanger*. www.ansonindustry.com: http://www.ansonindustry.com/pressure-vessel/u-tube-heat-exchanger.html
16. Jones D. R. et al. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications, 79*, 157-181.
17. Luyben, W. L. (2011). Heat-Exchanger Bypass Control. *American Chemical Society, 50*, 965-973.
18. Max S. Peters et al. (1991). *Plant Design and Economics for Chemical Engineers.* Singapore: McGraw-Hill.

19. McKay et al. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*, 239-245.

20. Pablo F. Navarrete et al. (2001). *Planning, Estimating, and Control of Chemical Construction Projects.* CRC Press.

21. Perry, R. H. (1999). Perry's Chemical Engineers' Handbook. McGraw-Hill.

22. Sahinidis, N. V. (2015). *BARON User Manual v.15.6.5.* Pittsburgh, USA: http://www.minlp.com/downloads/docs/baron manual.pdf.

23. Sahinidis et al. (2017). The ALAMO approach to machine learning. *Computers and Chemical Engineering, 106*, 785 - 795.

24. Sandro Salsa et al. (2013). *A Primer on PDEs - Models, Methods, SImulations.* Milano: Springer-Verlag Italia.

25. Serth, R. W. (2007). *Process Heat Transfer - Principles and Applications.* Burlington, USA: Elsevier.

26. Sobol, I. M. (1967). on of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychiclitel'noi Matematiki i matematicheskoi Fiziki, 7*, 784–802.

27. Sun Lin et al. (2013). Bypass selection for control of heat exchanger network. *Chinese Journal of Chemical Engineering, 21*, 276-284.

28. Ulrich, G. D. (1984). *A guide to Chemical ENgineering Process Design and Economics.* John Wiley and Sons.

29. Wild et al. (2008). ORBIT: optimization by radial basisfunction interpolation in trust-Regions. *SIAM Journal on Scientific Computing*, 3197-3219.

30. Yao et al. (2017). A Review of Industrial Heat Exchange Optimization. *Earth and Environmental Sciences* (p. 1-6). Sheffield, UK: IOP Publishing.

7 ACKNOWLEDGEMENTS

I want to thank all those people who have accompanied me throughout these long happy years at Politecnico di Milano and have shared their precious time with me, thus making my university experience amazing at all.

First of all, I wish to thank professor Flavio Manenti, who accepted me among his many students despite his innumerable commitments around the world: he is an unsurpassable person and has an infinite energy for everything he does. Besides, I really want to thank the doctorand Alessandro Di Pretoro, who followed me over more than one year throughout my whole thesis work and all the troubles we encountered on the way, but helping me every time as best as he could: actually, he became a friend for me after all this time spent together, after all the conversations and laughs. Furthermore, I want to thank my French professor Ludovic Montastruc of INP-ENSIACET of Toulouse, who made me start my thesis project by choosing the subject and guiding me during my initial steps of this work. Thanks to all my French colleagues, in particular Pierre and Lucas, fundamental for my settlement in my office, with their cups of tea and their funny conversations after lunch (at half past 11 a.m., damn!). More in general, I want to thank all professors, doctorands and people inside university who helped me in this work.

Afterward, I want to thank all my family. A very important thank is for my parents, Walter and Rita, who made everything possible without any doubt, with their love and their patience, by supporting me even if I'm quite annoying (I know it!); I hope they will be in my life for long, long time yet. Also, my brother Ottavio absolutely deserves to be thanked because he supported me a lot, with his affection, his wisdom and his useful advices for everything which always result to be true (damn!). For sure, my life would have been way more boring and more difficult without him.

Another person to be appointed is my cousin Massimo, who continuously tries to make me a better person by opening my mind, even if the way is still long, but he doesn't give up. I'm happy to drink some beer together sometimes and to talk about whatever. Of course, I cannot forget to name his beautiful children, Ginevra and Mattia, who always cheer the atmosphere wherever we are. I thank my uncle Bruno and my aunt Grazia (actually, Mariagrazia) too, for all the coffees after dinner offered to me and my mother.

I want to thank my uncle Edoardo and my aunt Tina, as they have been always present in our life. Also, my cousin Paola and her boyfriend Alberto must be thanked, as they donate me lots of splendid moments every time we meet.

Then, I want to thank my cousin Andrea, who is the best skiing companion ever, and his girlfriend Stefania with their beautiful curly child Alessandro, who is always scared as he sees me. Thanks to my uncle Ignazio and my aunt Rosa, who I often meet around in Bicocca. Furthermore, I thank my

twin uncles Mimmo and Franco too, with their formidable, serene lifestyle. Of course, I would like to thank my grandmother Carmela, even if she is gone more than one year ago: she has been an important pillar of our family.

Now, I want to thank all my friends: some of them have been constantly present in my life, some others have been beside me for some periods and others have entered my life in last years, but they have been all important to me.

First, I must thank the friends of my entire life, who I know since we were little adorable children: Pietro, with all his ambitions and his severe, but even right advices; Ame, with his incredible crinkly hair and those long conversations about the essence of life; Pana, who is proud of his mythic, modified red Audi and who donated me the best laughs I have ever had at all; Annina, with her fantastic stories and her psychological support that only she can give.

It is essential to me to thank Oreste and Laura, with their awesome dinners which allow us to spend carefree times altogether; their daughter Caterina, who always brings joy with her smile; also, I want to remember their missing son Filippo, who will be in our hearts forever.

I want to thank old friends, such as Leo, Ale and Federico, with whom I enjoyed countless nights together. For sure, I thank the crazy Save too, who acts like a tough guy, but he's got a very big heart. Do not forget to thank Silviona, who is always ready to hang out and make party, even if she talks too much. Thanks to Otta and Fede, always present for a spritz and never back home before 5 a.m. o' clock. Also, thanks to Cipo and Teo, the craziest and funniest couple you could ever meet on your way.

I cannot avoid thanking my old schoolmates of high school Manu, Anto, Ivo, Gulli, Nico, Ale, Teo, with whom I spent all my adolescence and I grew up during those important years, passing from being just a kid up to be a young man.

Then, I must thank all my university friends, who have shared with me joy and pain, laughter and tears, while studying together for the exams. Most of them have passed for that wonderful place called CdS (Casa dello Studente), where we all lived unforgettable moments. Thanks to my first classmates Filandro, Mattia, Samuel, Giallo and their relative consorts, with whom it looks like we are still 20-year guys at the beginning of university when we hang out. Thanks to Veronica, the first person I have encountered at Chemical Engineering, who helped me a lot not only for exams, but for everything. Thanks to Ebi and Jacopo, two crazy classmates with whom I prepared innumerable exams and spent a lot of wonderful nights. Thanks to Fuma, Volo, Mari, Denis, Sanaz, with whom I made many exams and I shared a lot of moments (and notes, of course!). Thanks to Vigna, Deme, Marco, Jacopino, Curz, Kaveh and Fra, inseparable buddies of CdS with a lot of hours of study together on our shoulders, but also many talks about everything which made those hours less stressing. Thanks to Nadia, Silvietta, Luciola and Ric, my classmates of misadventures with exams

and professors, but always smiling and comforting in times of need. Also, thanks to Nico, Martina, Giusy, Licia, Simona, Federico, Marghe and their relative consorts, friends of my Master Science, with whom I shared a lot of beautiful moments (and exams, of course!).
Of course, I want to thank all the friends I met during my Erasmus in Toulouse, who accompanied me throughout one of the best periods of my life, brief but very intense. Never forget all the memories and nights together! So, thanks to Tommi, Toni l'Eroe, Fabricia, Claretta, Alme, Er Tracina, Lo Sbirro, Luca il Cumpà, Chez Alessandro Rocco, Bea, Smarties, Gabri, Agne, Emanuela, Bright, Giulia, Elisa, Ele. Too many laughs together!

To all these incredible people, I say: THANK YOU! Really thank you for being an important part of my life!

Valerio Lorenzo Muci, Milan

April, 2019