Accepted Manuscript

Computers & Chemical Engineering

An International Journal of Computer Applications in Chemical Engineering

Title: Large scale optimization of a sour water stripping plant using surrogate models

Author: Natalia Quirante José A. Caballeroa

PII: S0098-1354(16)30135-1
DOI: http://dx.doi.org/doi:10.1016/j.compchemeng.2016.04.039
Reference: CACE 5458

To appear in: *Computers and Chemical Engineering*

Received date: 3-11-2015
Revised date: 26-4-2016
Accepted date: 26-4-2016

Please cite this article as: Quirante, Natalia., & Caballeroa, José A., Large scale optimization of a sour water stripping plant using surrogate models.*Computers and Chemical Engineering* http://dx.doi.org/10.1016/j.compchemeng.2016.04.039

Large scale optimization of a sour water stripping plant using surrogate models

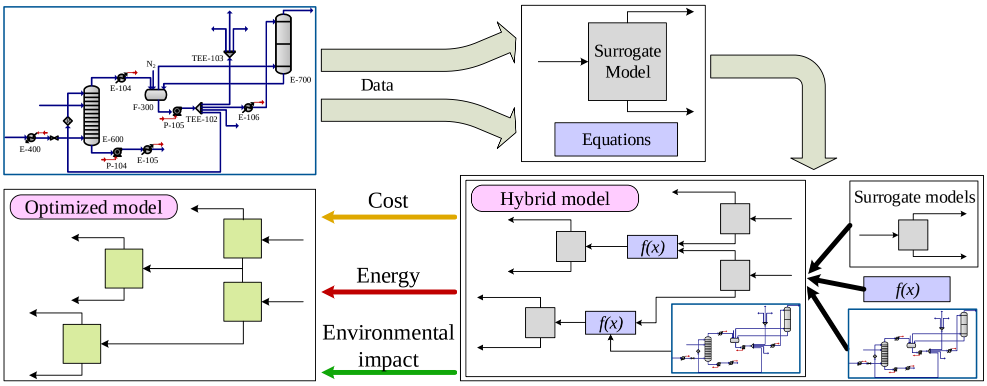
Natalia Quirante[a] and José A. Caballero[a,]*

Institute of Chemical Processes Engineering. University of Alicante, PO 99, E-03080 Alicante, Spain.

* Corresponding author: Tel: +34 965902322. Fax: +34 965903826.

E-mail addresses: natalia.quirante@ua.es (N. Quirante), caballer@ua.es (J.A. Caballero)

Graphical abstract



Highlights

- We develop an efficient multi-objective optimization method using surrogate models based on Kriging interpolation.
- We solve a hybrid simulation-optimization model formed by units of the flowsheet with low numerical noise, Kriging models and explicit equations.
- The hybrid approach (Kriging models, simulation and explicit equations) has proved to be robust and reliable.

Abstract

In this work, we propose a new methodology for the large scale optimization and process integration of complex chemical processes that have been simulated using modular chemical process simulators. Units with significant numerical noise or large CPU times are substituted by surrogate models based on Kriging interpolation. Using a degree of freedom analysis, some of those units can be aggregated into a single unit to reduce the complexity of the resulting model. As a result, we solve a hybrid simulation-optimization model formed by units in the original flowsheet, Kriging models, and explicit equations.

We present a case study of the optimization of a sour water stripping plant in which we simultaneously consider economics, heat integration and environmental impact using the ReCiPe indicator, which incorporates the recent advances made in Life Cycle Assessment (LCA).

The optimization strategy guarantees the convergence to a local optimum inside the tolerance of the numerical noise.

Keywords: process simulation, process optimization, Kriging interpolation, heat exchanger network, life cycle assessment.

Nomenclature

F_i	heat capacity flowrate of hot stream *i*
f_j	heat capacity flowrate of cold stream *j*
i	hot stream
j	cold stream
k	set of hot streams pinch candidates
l	set of cold streams pinch candidates
$Q_{k,i}^{hp}$	energy available from hot stream *i* above hot pinch candidate *k*
$Q_{l,i}^{cp}$	energy available from hot stream *i* above cold pinch candidate *l*
$q_{k,j}^{hp}$	energy required by cold stream *j* above hot pinch candidate *k*
$q_{l,j}^{cp}$	energy required by cold stream *j* above cold pinch candidate *l*
QC_j	energy to be transferred to cold stream *j*
QH_i	energy to be transferred from hot stream *i*
Q_S	heat supplied by the hot utility
Q_W	heat removed by the cold utility
T_i^{in}	inlet temperature for the hot stream *i*
T_i^{out}	outlet temperature for the hot stream *i*
t_j^{in}	inlet temperature for the cold stream *j*
t_j^{out}	outlet temperature for the cold stream *j*
T_k^{in}	inlet temperature for a pinch candidate *k*

t_l^{in} inlet temperature for a pinch candidate *l*

WC_j^{Above} binary variable. Represents the case when the stream *j* lies above the pinch candidate

WC_j^{Below} binary variable. Represents the case when the stream *j* lies below the pinch candidate

WC_j^{Middle} binary variable. Represents the case when the stream *j* crosses the pinch candidate

WH_i^{Above} binary variable. Represents the case when the stream *i* lies above the pinch candidate

WH_i^{Below} binary variable. Represents the case when the stream *i* lies below the pinch candidate

WH_i^{Middle} binary variable. Represents the case when the stream *i* crosses the pinch candidate

YC_i^{Above} binary variable. Represents that hot stream *i* lies above the temperature of the cold pinch candidate plus ΔT_{min}

YC_i^{Below} binary variable. Represents that hot stream *i* lies below the temperature of the cold pinch candidate plus ΔT_{min}

YC_i^{Middle} binary variable. Represents that hot stream *i* crosses the temperature of the cold pinch candidate plus ΔT_{min}

YH_i^{Above} binary variable. Represents that hot stream *i* lies above the temperature of the hot pinch candidate

YH_i^{Below} binary variable. Represents that hot stream *i* lies below the temperature of the hot pinch candidate

YH_i^{Middle} binary variable. Represents that hot stream *i* crosses the temperature of the hot pinch candidate

ZC_j^{Above} binary variable. Represents that cold stream *j* lies above the temperature of the cold pinch candidate

ZC_j^{Below} binary variable. Represents that cold stream *j* lies below the temperature of the cold pinch candidate

ZC_j^{Middle} binary variable. Represents that cold stream *j* crosses the temperature of the cold pinch candidate

ZH_j^{Above} binary variable. Represents that cold stream *j* lies above the temperature of the hot pinch candidate minus ΔT_{min}

ZH_j^{Below} binary variable. Represents that cold stream *j* lies below the temperature of the hot pinch candidate minus ΔT_{min}

ZH_j^{Middle} binary variable. Represents that cold stream *j* crosses the temperature of the hot pinch candidate minus ΔT_{min}

ΔT_{min} heat recovery approach temperature

1. Introduction

The simulation of a chemical plant can be represented by a large system of linear and nonlinear equations of the form:

$$f(x) = 0 \tag{1}$$

Where *f* is a vector of functions and *x* is a vector of variables. The variables represent composition, temperatures, pressures, flow rates, etc. and the functions are obtained from conservation of mass and energy, chemical equilibrium, kinetics and transport phenomena or physical properties calculations. Even a small chemical plant can involve thousands of equations and variables. In some cases, it is possible to write and solve the complete set of equations by using an adequate modeling system (e.g. ASCEND (Piela et al., 1991) or gPROMS (Process Systems Enterprise, 2000)). Those modeling systems include databases of chemical and thermodynamic properties and robust numerical methods. However, as the model becomes more complex the convergence is more difficult and the possibility of physically meaningless solutions increases. In those cases, good initial points and/or complex initialization strategies are mandatory.

Alternatively, instead of solving all the equations simultaneously, it is possible to use a modular approach. In this case, a given module is solved using specific numerical methods, including their initialization strategies. To converge the entire flowsheet all units are solved following a pre-specified sequence. In order to assemble the different modules, the output from one module is used as input for the next one so that the information flow matches the material flow (a notable exception is the process simulator Aspen-Hysys (Hyprotech, 1995 - 2011)). If there are recycles in the flowsheet we must also select a set of “tear streams” (or variables) and iterate over these variables by repeatedly solving the entire flowsheet. Mathematically the original problem is rewritten as follows:

$$\begin{aligned}
&x_i^{out} = g_i\left(x_i^{in}, u_i\right) \quad i = 1 \ldots n_u \\
&x_i^{out} = x_j^{in} \quad output\ from\ i\ es\ an\ input\ to\ j \\
&\left\| x_{tear}^{k+1} - x_{tear}^{k} \right\| \le eps
\end{aligned} \tag{2}$$

Where *u* is a set of module specific parameters, x^{out}, x^{in} are a subset of variables related to the input and output of module *i*. Outputs from module *i* become the inputs of the following module(s) in the flowsheet. The set of tear variables (related to streams or other user added convergence blocks) must also be explicitly included in the model.

The theory related to the simulation and convergence of sequential modular flowsheets was developed in the 1970s. A good overview can be found in the book by Westerberg et al. (1979).

Some noteworthy advantages of sequential modular simulators are: a) Different modules can be developed and tested independently. b) Solution methods can be tailored for each model, independently of its final use. c) Data can be easily checked for completeness and consistency. d) New modules can be easily added. Due to these advantages, it is not surprising that modular simulators are the dominant approach.

However, when we move from simulation to optimization modular simulation based approach loses some of its advantages:

- The selection of independent variables is constrained by the rigid input-output structure.
- Most of the modules are in the form of «gray box models» in which the final user has not access to the explicit equations; therefore, derivative information must be estimated using perturbations of independent variables by any finite difference scheme.
- Implicit modules inherently include some numerical noise. If the noise is low enough the unit can be used in an optimization model, provided that the convergence is fast enough. This is usually the case of a single flash, mixers, splitters, compressors or pumps, among other units in a flowsheet. However, even in a slightly noisy model, an accurate estimation of a derivative is not possible, resulting in unexpected behavior of NLP solvers. Of course, second derivatives are usually not even considered and some NLP solvers (i.e. CONOPT (Drud, 1996)) base part of their performance on accurate second derivatives.
- Recycles act as noise amplifiers, further increasing the numerical problem.
- A module should converge in a relatively short CPU time. Each time that the solver needs a new gradient, a given module must be solved at least twice for each independent variable affecting that module. If

the convergence is not fast enough the total CPU time could become prohibitive.

- The lack of convergence of a given module in any moment of the optimization crashes the entire optimization. It is possible to develop strategies to recover from simulation convergence failures, but the behavior after a recovery is solver dependent and not always reliable.

To overcome all these drawbacks, in recent years, surrogate models have been proposed as an alternative to process models which have a modular structure, because they ensure an acceptable degree of accuracy and they are computationally efficient (Chung et al., 2011). In this context a metamodel or data driven surrogate model –for simplicity “surrogate models” hereinafter- is a relatively simple combination of mathematical functions, based on data generated from the simulation with the sole purpose of approximating the input-output relationship of the simulation. While the original simulation model could be difficult to solve, noisy or time consuming, the surrogate model must be relatively easy to solve and noise free (Palmer & Realff, 2002).

In the optimization field, surrogate models have become popular due to their applicability (Caballero & Grossmann, 2008; Queipo et al., 2005; Wang & Shan, 2006) and we can differentiate two approaches. The first locates the most relevant variables of the entire flowsheet through a sensitivity analysis and then generates a surrogate model based on these variables. If the number of variables is large, then the number of sampling points must also be large to capture the behavior of the original model and/or a frequent resampling is usually needed through the optimization algorithm. The second approach disaggregates the simulation model into different blocks and each block is modeled separately before optimization. This ensures that smaller and more robust models are generated (Cozad et al., 2014). The disaggregated process units can be linked by the variable connectivity to formulate complex optimization models (Cozad et al., 2014).

The HEN is a basic component in many industrial processes because they are responsible for large amounts of energy consumption (Allen et al., 2009). For this reason, research in the area of HEN synthesis has been developed with considerable effect on the industry (Al-mutairi, 2010; K. F. Huang & Karimi, 2013).

Additionally, reduction of energy consumption can achieve the minimization of environmental impacts (Lara et al., 2013; Morar & Agachi, 2010).

In this paper we deal with the optimization of a large scale actual sour water stripping plant (SWS) with the following relevant characteristics:

- We use a novel approach in which some parts of the plant are substituted by Kriging models (in particular stripping columns), some units are maintained in the process simulator (those that do not introduce numerical noise like pumps or heat exchangers), and parts of the model are defined in terms of explicit equations, in particular, all the equations related to heat integration and Life Cycle Assessment.
- We do not follow a complete distributed approach (where each piece of equipment is substituted by a Kriging model) nor a global one (where the complete flowsheet is substituted by a surrogate model), instead we use an analysis based on feasibility and degree of freedom considerations that allows aggregating some equipment in a single and more robust surrogate model.
- We simultaneously perform the optimization of the operating conditions of the flowsheet and the heat integration using the pinch location method (added to the model in form of explicit equations with continuous and binary variables). As far as we know, this kind of optimization has been previously done only in equation based systems involving a reduced number of streams (around 3 hot and 3 cold streams at most) and of course no on a very large scale model.
- Convergence of the recycle streams is carried out by the optimization solver (and not by the simulation) by transforming the convergence blocks to explicit equations avoiding inefficient and time consuming iterations.

The result is a reliable and robust optimization model.

In the rest of the paper, we first discuss the practical implementation and the optimization algorithm. Then we introduce the optimization of an actual stripping plant located in Germany. First, we perform the optimization of the stripping plant, minimizing the operating costs without considering heat integration, and evaluate the environmental performance through a Life

Cycle Assessment (LCA). Then we introduce the heat integration and repeat the optimization together with the LCA. And finally, we present a broad discussion through the case study.

2. Methods

In this work, we focus on the Kriging (Krige, 1951) interpolation to approximate models. Kriging metamodels combine relatively small sampling data with computational efficiency. Usually, data obtained from larger experimental areas are used to fit Kriging models. Therefore, Kriging models have been used for sensitivity analysis and optimization (Kleijnen, 2009).

Important studies have been performed with Kriging models by disaggregating parts of the model (Caballero & Grossmann, 2008) or using the full system approach (Davis & Ierapetritou, 2007; D. Huang et al., 2006; Palmer & Realff, 2002). An interesting summary of Kriging simulation applications can be found in the review by Kleijnen (2009). Caballero and Grossmann (2008) studied modular flowsheet (disaggregated) optimization using Kriging models to represent process units with low-level noise. Complete process Kriging models were used by Davis and Ierapetritou (2007) to find global model solutions and later refine them using local response surface around the optima. The optimization of steady-state simulators using surrogate models was studied by Palmer and Realff (2002). To deal with uncertainty in black-box systems, D. Huang et al. (2006) used Kriging models on complete processes. Henao and Maravelias (2011) employed disaggregated models for each unit in a flowsheet using artificial neural networks. Quirante et al. (2015) used Kriging interpolation for the rigorous design of distillation columns and distillation sequences, explicitly including integer variables in the surrogate model.

In this paper, we follow a disaggregated approach, but instead of using a surrogate model for each unit in the flowsheet, we substitute only those modules that could potentially introduce numerical problems in the optimization. The rest of the units: phase separators, mixers, splitters, heaters, coolers, pumps, etc. are maintained in their original form. In this way we have a hybrid system that can simultaneously deal with:

- Modules at the level of process simulator.
- Third party modules developed in any other simulation environment.
- Explicit equations. This could be a unit operation added in equation form or constraints added by the designer.

Different authors have proposed different general procedures for the creation and use of surrogate models (Palmer & Realff, 2002; Welch & Sacks, 1991). All of them share the main basic steps with different modifications depending on the final objective (local or global optimization), the availability of derivatives and the accuracy of the initial Kriging interpolator. Biegler et al. (2014) in the context of multi-scale optimization, proposes three algorithms for using surrogate models with trust regions concept from non-linear programming that guarantee convergence to the optimum of the original problem. Biegler et al. (2014) also established the convergence conditions of these algorithms. In this paper, we follow an exhaustive sampling to minimize the number of resamplings and Kriging calibration (algorithm 3 in the Biegler's taxonomy). It is worth remark that this approach is only feasible when the number of degrees of freedom in each Kriging model is small (say no more than 5 or 6 degrees of freedom). The disaggregation strategy that we follow in this work generates a relatively large number of Kriging models with reduced number of degrees of freedom.

To develop a robust and convergent trust region algorithm involving surrogate models, the following conditions must hold (Conn et al., 2000):

For the original model:

1. Functions must be twice continuously differentiable on R^n.
2. Functions are bounded below for all variables in their domain.
3. The second derivatives are uniformly bounded for all the variables in the domain.

For the surrogate model:

4. At each iteration, the surrogate model is twice differentiable inside the trust region.
5. The values of the original and surrogate models coincide in the current iterate inside the trust region.

6. The gradients of the original and surrogate models coincide, for every iteration, inside the trust region.
7. The second derivatives of the surrogate models remain bounded within the trust regions.

Conditions for the original model can, in general, be ensured because models behind a process simulator are based on material and energy balances; heat, mass or momentum transfer equations, equilibrium relations, etc. All these equations are continuous, differentiable and bounded. However, some care must be taken when used in a black box model. For example, different equations can be used to estimate the heat transfer coefficients depending on the flow regimen. The set of equations is also different in the flash calculation of a single phase or if multiple phases appear. The designer must be aware of these situations and capture this behavior.

In the case of surrogate models conditions 4 and 7 can be guaranteed by the surrogate construction. Condition 5 can be ensured by constructing accurate surrogates over the trust region. However, gradients of the original and surrogate models could differ. A common approach to solve this difference consists of using scaled functions by using local corrections to the current iteration (Agarwal & Biegler, 2013).

$$\begin{aligned}\widetilde{\Phi}_k^S(x) = \Phi_k^S(x) + \left(\Phi(x_k) - \Phi_k^S(x_k)\right) \\ + \left(\nabla\Phi(x_k) - \nabla\Phi_k^S(x_k)\right)^T (x - x_k)\end{aligned} \tag{3}$$

Where $\widetilde{\Phi}_k^S(\cdot)$ is the corrected (scaled) surrogate model at the current iteration, $\Phi_k^S(\cdot)$ is the uncorrected surrogate model and $\Phi(\cdot)$ is the original model. However, in a noisy model it is not possible to calculate the derivative of the model; in fact this is one of the main reasons to use a surrogate model. To circumvent this problem, Quirante et al. (2015) proposed including a matching gradient step that basically consists of contracting the trust region around the optimal solution obtained in the previous step (note that resampling is needed) and re-optimizing starting from the optimal solution before contraction. In a noisy system, we must finish when there is no improvement in two consecutive contractions in a small, but large enough (to avoid adjusting the noise) region.

A critical aspect in surrogate modeling is the selection of the sampling points. This point cannot be randomly selected but a pre-specified space filling design must be used. Biegler et al. (2014) showed that frequent resampling of the original models can result in prohibitively large computational times. Instead, they proposed exhaustive evaluations of the original models over large trust regions before starting the optimization. With sufficiently accurate surrogate models it is possible to minimize (or even avoid) resampling during the optimization. Of course, there is a tradeoff between the cost of an a-priori sampling and the cost of some intermediate re-sampling. However, this tradeoff is case dependent. In this case, taking into account that each resampling also involves a Kriging calibration we will try to minimize resampling as much as possible by performing exhaustive a-priori sampling, even though we recognize that maybe this is not the optimal strategy.

In order to get good Kriging models with reduced initial error while minimizing the necessity of resampling and recalibrating, a correct distribution of sampling points is mandatory. The final quality of the Kriging model depends more on the uniformity of the sampling distribution than on its randomness. If we use a set of points randomly distributed without any other consideration, we could expect surrogate models with bad performance (independently on the surrogate model). If the model includes some measure of the quality of the parameters like confidence intervals in the case of regression models or estimated variance in the case of Kriging we would expect to obtain large values of these estimators if the sampling points are not correctly selected (Diwekar, 2003). There are different variance reduction techniques like Hammersley, Halton or Sobol sequences, Latin Hypercube sampling, etc. A discussion on sampling can be found in the work by Sasena (2002).

In this paper we select the ‘maxmin’ approach to distribute the sampling points; we maximize the minimum distance between two sample points in a normalized space (all variables range between 0 and 1). However, instead of distributing N points following the ‘maxmin’ approach, we fix 2^D points to the D-dimensional vertex of the hypercube that forms the feasible space and then we distribute the rest ($N-2^D$) points following the ‘maxmin’ approach. In this way, we ensure that Kriging does not perform ‘extrapolations’ near the vertices of the feasible region.

It is worth mentioning that deterministic optimization methods, like the approach proposed in this paper, are not the only alternative for dealing with these problems.

Stochastic methods have proved to be a good alternative for solving hybrid simulation–based optimization problems. Derivative-free optimization (DFO) is a class of algorithms designed to solve optimization problems when derivatives are unavailable, unreliable or prohibitively expensive to evaluate. Although there is a vast literature on metaheuristic optimization, combination with chemical process simulators is a relatively recent development (Dantus & High, 1999; Eslick & Miller, 2011; Gutiérrez-Antonio & Briones-Ramírez, 2009; Gutiérrez-Antonio et al., 2011; Leboreiro & Acevedo, 2004; Torres et al., 2013). Although DFO algorithms can be used in models with costly and/or noisy function evaluations, these methods are often constrained to models in which the number of degrees of freedom does not exceed about 10 (Rios & Sahinidis, 2013).

On the other hand, energy efficiency is a crucial aspect of chemical processes. One of the main reasons to develop techniques for efficient and sustainable energy use is the increasing global demand, related to the high cost of energy due to the quick decrease in the availability of fossil fuels, the technological barriers and prohibitive prices of renewable energy, and the strict standards that regulate carbon dioxide emissions, to mitigate the greenhouse effect and its consequences (Gharaie et al., 2013; Hasan et al., 2010; K. F. Huang & Karimi, 2013; Razib et al., 2012; Wechsung et al., 2011). Additionally, the most effective method to reduce costs is the use of energy from process streams through thermal integration between heat exchangers and cooling and/or heating systems. The optimal Heat Exchanger Network (HEN) is through the thermal integration of the system.

3. Algorithm implementation

Surrogate models based on Kriging interpolation combine computational efficiency with relatively small sampling data compared to other methods of approximating a model. For example, regression by splines usually requires moderate data sets (Friedman, 1991) while neural networks usually require large data sets (Himmelblau, 2000).

Kriging was developed by the South African mining engineer Daniel G. Krige in his Master Thesis (Krige, 1951).

The Kriging fitting is composed of two parts: a polynomial expression and a deviation from that polynomial:

$$y(x) = f(x) + Z(x) \tag{4}$$

where *Z(x)* is a stochastic Gaussian process that represents the uncertainty about the mean of *y(x)* with expected value zero. The covariance for two points x_i and x_j is given by a scale factor σ^2 that can be fitted to the data and by a spatial correlation function $R(x_i,x_j)$. The most common alternative for the spatial correlation function in Kriging models is to use the extended exponential (Sacks et al., 1989).

$$\begin{aligned} R(x_i, x_j) &= exp\left(-\sum_{l=1}^{d} \theta_l \left(x_{i,l} - x_{j,l}\right)^{P_l}\right) \\ &= \prod_{l=1}^{d} exp\left(-\theta_l \left(x_{i,l} - x_{j,l}\right)^{P_l}\right) \end{aligned} \tag{5}$$

where $\theta_l \geq 0$ and $0 \leq P_l \leq 2$ are adjustable parameters. The value of θ_l shows how fast the correlation goes to zero as we move in a *l*th coordinate direction. The parameter P_l determines the smoothness of the function which is usually fixed to 2 (Gaussian Kriging) in all coordinates.

In Kriging fitting, when a function is smooth, the degrees of the polynomial *f(x)* does not affect significantly the resulting metamodel fit because *Z(x)* captures the most significant behavior of the function. This is an important advantage of Kriging models. Usually, a simple constant term (μ) is enough for a good prediction.

A comprehensive description of all the details of Kriging interpolation can be found in references (Jones, 2001; Jones et al., 1998; Palmer & Realff, 2002; Quirante et al., 2015).

Before presenting a detailed description of the algorithm it is interesting to introduce the characteristics of the problem we are dealing with.

The starting point is a complex flowsheet that is usually defined in a process simulator. Some general characteristics of the flowsheet are:

- It can contain «gray box» units defined in the database of the process simulator. The specific equations, and of course their derivatives, are usually closed to the final user.
- Some unit operations can be defined by third party modules. For example, proprietary process units. For this kind of models we have all the possibilities: a) Modules without access to the code. In this case, we can consider the units equivalent to any other module in the flowsheet. b) Modules with access to the code. In this case, it is possible to get exact derivatives (inside the computer precision) by using automatic differentiation. c) The model is in equation form. In this case, we have two options; either we can maintain the module identity by solving the equations at each iteration, or explicitly include the equations in the whole model and solve those equations together with the rest of the flowsheet.
- The flowsheet could include important "recycle of information" either by recycle streams (identified as tear streams in the flowsheet) or any other convergence blocks (these blocks have different names depending on the process simulator, i.e. Adjust in Aspen-Hysys (Hyprotech, 1995 - 2011), Forward or Backward controllers in ChemCad (Chemstations, 2012), Specifications in Aspen-Plus (Aspen Technology, 1994-2015)).

We are interested in performing an efficient optimization but at the same time maintaining the rigor of a process simulator. As mentioned before, there are three reasons why an NLP solver presents bad performance or fails when is directly interfaced with a process simulator: Large CPU execution times in some unit operations, numerical noise and lack of convergence. To overcome these problems we substitute these "badly behaved" models by surrogate models based on the Kriging interpolation. In this case, we follow a disaggregated approach in which only those units or modules that could produce numerical problems are substituted by Kriging surrogates, the rest of the units are maintained in the simulator.

Recycles, either stream recycles or convergence blocks, introduce two numerical problems. In the first place, they can act as noise amplifiers because small errors can be propagated through the cycle and secondly, the CPU time to converge the flowsheet could be large because all units must be converged inside the recycles at each flowsheet iteration. Instead, we let the NLP solver converge all the recycles: We explicitly include all the recycles and convergence blocks as constraints in the NLP model. With this approach, we gain in speed and reliability and completely avoid the numerical problems mentioned above.

Finally, it is possible to add any given model in equation form (e.g. equations for energy integration or LCA) or any set of constraints or bounds to the model in the same way as in a regular NLP model.

It is worth remarking that the Kriging interpolation does not accept cross-correlation between different simulation outputs, and univariate Kriging models are fitted. In other words, we can adjust multiple inputs and single output models so a given multiple-output model will require multiple Kriging interpolators. There are, however, cokriging methods that take advantage of the covariance between two or more variables that are related, and are appropriate when the main attribute of interest is sparse, but related secondary information is abundant. Cokriging requires the same conditions to be satisfied as Kriging does, but demands more modeling, and computation time. The common cokriging methods are multivariate extensions of the Kriging system of equations and use two or more additional attributes. In our case, even though some variables are clearly correlated, independent variables are enough to define the problem and we do not expect better numerical performance but much more computational effort. However, conservation properties (mass and energy balances) must be rigorously maintained (e.g. we cannot adjust all the flows of all the components because small errors could violate the mass balances). As a consequence, we select a set of output variables (those with the largest sensitivity) and calculate the rest through conservation balances.

The model we are dealing with is, therefore, a hybrid model: explicit units in process simulator, multiple Kriging modules, third party modules connected to the simulation and explicit equations. Conceptually the model can be written as follows:

$$
\begin{array}{ll}
min: f(x) & \\
s.a.\ x_{i,j}^{Out} & \\
= G_{i,j}^{S}\left(x_i^{In}, u_i\right) & i \in Kriging\ Unit;\ \ j \in Kriging\ model_i \\
\quad x_k^{Out} = G(x_k^{In}, u_k) & k \in Unit\ in\ process\ simulator \\
\quad x_j^{Out} = x_i^{In} & connectivity
\end{array}
\qquad (6)
$$

$$\left\| x_{tear}^{sup} - x_{tear}^{calc} \right\| \leq eps$$
$$h(x) = 0$$
$$g(x) \leq 0$$

Where $f(\cdot)$ is the objective function. The first constraint represents the input-output structure of the Kriging interpolators. The second constraint is the input-output structure for the units in process simulators or third party models. The third constraint is related to the connectivity equations between different units in the flowsheet. These equations can be explicitly included in the form of equations or implicitly by just propagating the information through the flowsheet. We follow this second approach. The fourth constraint transfers the recycle structure of the flowsheet to the NLP solver. Finally, the last two constraints are explicit equations added to the model.

With all the previous comments in mind, the algorithm for solving these models is as follows (a scheme of the flowchart has been included in Fig. 1):

< Insert Fig. 1 >

Fig. 1. Flowchart of Kriging based NLP optimization algorithm.

1. The starting point is a converged, large scale, flowsheet. Usually, a given flowsheet includes specifications (like purities, recoveries, etc.) that must be met but could be difficult to converge. In these cases, the first step consists of locating these difficult specifications and substituting them by easier to converge specifications and transferring the difficult to converge constraints to the NLP solver in the form of explicit equations. For example, in a conventional distillation column with known pressure (two degrees of freedom) concentration specifications are much more difficult to converge than distillate flowrates and reflux ratio. Some process simulators (Aspen Plus) follow this approach in distillation columns.
2. A sensitivity analysis for each unit in the flowsheet will provide the following information: a) Which variables are the most important in the flowsheet and which can be neglected without affecting the optimization performance. b) Which units introduce unacceptable numerical noise and must be substituted by Kriging interpolators and which can be

maintained in the process simulators. c) Indirectly, the CPU time to converge a given unit. If it is too large then we must consider the possibility of using a surrogate model with the unit. Finally, at this point, we must consider the possibility of merging two or more units in a single surrogate model. For example, two or more columns connected by a thermal couple with the objective of reducing the number of degrees of freedom of the surrogate model and increasing the robustness of the optimization.

3. All the recycle information is removed from the flowsheet and transformed in explicit equations in the solver. In this way, the NLP solver will 'see' an acyclic system avoiding both unnecessary iterations and numerical noise amplification.

4. Sampling. In this paper, we select the 'maxmin' approach to distribute a set of N points maximizing the minimum distance between two sample points in a normalized space. Note however, that 2^D of those points (D is the number of independent variables) are fixed to the corners of the D-dimensional hypercube that define the domain of the independent variables. In this way, we ensure that Kriging is not making extrapolations. A detailed description of the maxmin procedure is included in Appendix B.

5. For a given trust region, ideally the complete domain for every surrogate model, we calibrate the Kriging models and validate the accuracy of the model by two approaches. The first is cross-validation: a point is removed and its value re-evaluated with the rest of the points. This procedure is repeated with all the sampling points. And, second we use a minimum set of 100 randomly selected points. The consideration of an error as «small» is case dependent. Depends on the sensitivity of the variable in relation to the rest of the model. As commented above, one of the first steps in the algorithm consists of performing a sensitivity analysis to determine what the most relevant variables are. From this analysis, we also know what could be the effect of a given error. Roughly speaking, for this case of study, the maximum error is no larger than 5%. In any case, the final point must be checked against the process simulator.

6. If the error is small enough, the Kriging surrogates can be used to substitute the original one. If this is not the case we can increase the

number of sampling points or reduce the trust-region. Again we have a tradeoff: a very large number of sampled points increase the CPU time for calibrating the Kriging that is itself an NLP problem. The bottleneck is the time required to invert the correlation matrix and the time needed to perform an interpolation. Also, small trust regions could eventually require resampling and recalibrating the Kriging models. As commented before, in this paper we follow an exhaustive sampling to minimize the number of resamplings and Kriging calibrations (algorithm 3 in the Biegler's taxonomy (Biegler et al., 2014)). In the case in which the model cannot be considered accurate enough, a reduction of the domain is mandatory. In this case, we would follow a trust region approach following the algorithm presented by Caballero and Grossmann (2008).

7. Solve the model given in Eq.(6). Note that for all units calculated directly from the flowsheet, derivative information is calculated by a finite difference scheme. However, if the numerical noise is small and the convergence is fast, this approach is satisfactory. For the rest of the model, derivatives are calculated either by automatic differentiation or even symbolically.
8. The optimal solution of step 6 is not necessarily the optimal solution of the original model. If the original trust region does not cover the complete domain and the solution is in a boundary of the region, we must resample around the optimal point and repeat step 6. Even if we have an interior point, we cannot guarantee that the Karush-Kuhn-Tucker (KKT) point of problem 6 is also the optimum of the original model, because we cannot guarantee the gradient matching property. In this case, we must perform a 'gradient matching' step by contracting the trust region around the optimal solution (resampling) and repeat step 6. If both solutions are inside a tolerance then the optimization finishes, if not we must contract the original trust region and repeat again from step 6. It is important to remark that, in any case, the sampling points must be separated enough to avoid adjusting the numerical noise.

4. Case study: sour water stripping plant

In the petroleum refining industry, large volumes of water are used and large volumes of wastewater are produced (Joint Research Centre, 2013). Water is often in direct contact with some process streams (i.e. oil), therefore, it is very contaminated. This contaminated water is called sour water. The two most prevalent pollutants found in sour water are H_2S and NH_3, resulting from the destruction of organic sulfur and nitrogen compounds during desulfuration, denitrification, and hydrotreating.

The aim of sour water treatment is to remove sulfides and ammonia from the water. There are several technologies for sour water treatments, stripping with steam or flue gas, air oxidation to convert sulfides to thiosulfates, or vaporization and incineration. In this work we have considered that the sour water is stripped by steam.

The case study corresponds to the Sour Water Stripping (SWS) plant of a refinery located in Germany. The flowsheet was obtained from a work by Torres et al. (2013). This plant treats sour water coming from four different sources: an oil vacuum distillation unit, a fluid catalytic cracker and an amine regeneration fractionator, a crude distillation unit and a petrochemical complex with content in ethanol and ethyl tert-butyl ether (ETBE).

The stream with ethanol and ETBE is sent to a stripper, where it is stripped with steam, recovering an ethanol-rich gas that is dispatched to the Fluid Catalytic Cracker (FCC) unit.

The remaining water is sent to flash drums, where vapor and liquid hydrocarbons are removed. After the flash drums, the sour water is sent to the first set of strippers (strippers E1, E2 and E3 in Fig. 2) where, in contact with steam, ammonia and H_2S are removed. The aim of the SWS plant is not only to remove pollutants from water but also to achieve a high recovery of ammonia and hydrogen sulfide, separately. Then, overhead streams are mixed and sent to a second stage, where a high purity of hydrogen sulfide is recovered by overhead (units E4, E5 and E6 in Fig. 2). The bottom, which contains the ammonia is sent to the ammonia stripper (units E7 and E8 in Fig. 2) where ammonia-rich gas is recovered (79% w/w). The sulfides and ammonia free-water streams are split to be recycled with the feed streams, to be reused in other refinery processes and to be sent to the flare (unit E9 in Fig. 2).

< Insert Fig. 2 >

Fig. 2. Complete flowsheet of the sour water stripping plant. The flowsheet has been obtained from a work by Torres et al. (2013) where the heat exchanger network has been removed.

Two different property packages are used in the simulation; the NRTL model is used in streams and units involving ethanol and ETBE (streams S27-S35 and unit E3 in Fig. 2) and the SourPR model is applied to the rest of the model.

The main objective is to optimize the SWS plant operating conditions, but changing the flows and temperatures of some streams directly affects the energy integration of the plant. Therefore, we will also redesign the heat exchanger network. In order to avoid a pre-specified heat exchanger configuration that could be inefficient, the first step consists of removing all heat exchangers in the flowsheet and substituting them by simple heaters and coolers. The development of an optimal heat exchanger network involving a relatively large number of process streams (in this case study there are 7 hot and 4 cool streams) is a challenging problem by itself. The simultaneous optimization and heat integration based on a superstructure approach (see for example Yee and Grossmann (1990)) results in a very large highly non-convex Mixed Integer non-Linear Programming (MINLP) problem. Instead, taking advantage of the fact that energy is the dominant cost in the HEN, we simultaneously optimize the energy consumption, for a given minimum approach temperature and the operating conditions, and then we design the optimal HEN.

For fixed values of heat flows and input and output temperatures for all the streams involved in the heat exchanger design, the minimum utility consumption can be calculated either using the classical "Tableau" approach proposed by Linnhoff and Flower (1978) or using the transshipment problem (Papoulias & Grossmann, 1983). However, these approaches rely on the temperature interval concept. If the input and/or output temperatures are not fixed the temperature intervals could change. This is equivalent to introduce discontinuities and non-differentiabilities in the model. To overcome this difficulty, as far as we know, there are two alternatives; the first is the "*pinch location method*", proposed by Duran and Grossmann (1986) and reformulated as a disjunctive problem by Grossmann et al. (1998). The second is an implicit enumeration approach proposed by Navarro-Amorós et al. (2013). Both approaches have similar numerical performance, however, the disjunctive version of the pinch location method (Grossmann et al., 1998) generates smaller size models, so in this paper we follow this approach. An overview of the pinch location method in its disjunctive formulation has been included in Appendix A.

In order to avoid, as much as possible, getting trapped in a local optimum, we first optimize the flowsheet without taking into account heat integration. This intermediate step is used as the initial point of the complete model including heat integration and LCA analysis.

All relevant data related to the input streams and equipment characteristics are included in Table 1.

Specifications for output streams are in Table 2. All these specifications are transferred to the NLP model as constraints.

The second step (according to the algorithm implementation section) consists of performing a sensitivity analysis to determine which units must be maintained in their original form in the flowsheet, which units must be substituted by Kriging surrogate models and if it is convenient or not to merge some units in a single surrogate model. There are three main criteria to decide whether to substitute a unit (or set of units) by a surrogate model: large CPU convergence times, unacceptable numerical noise, and lack of convergence in the complete space of the domain. The convergence is fast enough for all the units. However, all the stripping columns are slightly noisy and convergence of some units (E4, E5, E7 and E8) eventually requires good initial points. The rest of the unit operations, phase separators, pumps, valves, heater or coolers, are maintained in their original form in the process simulator.

If we substitute each stripping column in the original model by a Kriging surrogate then some surrogate models have a large number of degrees of freedom. In particular, units E4 and E5 form a highly integrated system with a thermal couple (liquid stream from E4 to E5 and vapor stream from E5 to E4) and a recycle stream. Therefore, it is numerically more efficient to merge these two columns in a single surrogate. A similar situation appears with stripping columns E7 and E8. A scheme of the resulting flowsheet is shown in Fig. 3.

< Insert Fig. 3 >

Fig. 3. Simplified process flowsheet of the SWS plant. Note that unit ED is the result of merging the original columns E4 and E5 in a single surrogate

model, and unit EC is the result of merging the original columns E7 and E8 in a single surrogate model. For the sake of clarity, in the scheme, pumps, valves, coolers and heaters have been removed.

The next step consists of removing all the "recycles" from the flowsheet and transferring this information to the NLP solver. As previously commented, letting the NLP solver converge the recycle information is numerically much more efficient than converging the complete flowsheet at each iteration. The original flowsheet contains 9 recycle streams. Three of these recycle streams form part of the integrated system E4-E5 and E7-E8. The rest must be explicitly transferred in the form of constraints to the NLP solver. Taking into account that each stream has c+2 degrees of freedom (c is the number of components in the stream), we explicitly add 30 constraints to the NLP model.

The efficiency of the stripping depends on the steam flow rate, feed composition, and temperature, number of trays and feed location. In this paper we have a fixed structure; therefore, we have only considered the steam flow rate, feed composition and feed temperature as variables. The maximum concentration of NH_3 and H_2S was fixed according to the legal limits of the industrial emissions of pollutants (see Table 2).

One of the advantages of using a disaggregated approach is that, instead of using a single surrogate with a large number of independent variables, we calibrate a set of smaller surrogate models. If the number of degrees of freedom is not too large it is possible to follow the strategy proposed by Biegler et al. (2014), and perform exhaustive sampling a priori in order to minimize resampling and recalibration. Table 3 shows all the input-output Kriging models used in this example. Fig. 4 shows, as a typical example, the results from cross-validation, and Table 4 shows the parameters of all the Kriging models. The relatively low errors of all the Kriging surrogate models in the variables domain, allow us to use these models instead of the original ones. In any case, the final contraction step in order to ensure a KKT point is always performed.

< Insert Fig. 4 >

Fig. 4. Cross-validation for vapor obtained from unit ED. (a) mass flow H_2S, (b) mass flow NH_3, and (c) mass flow H_2O.

The first aim of this work is to minimize the costs of the SWS plant. As we are working with an existing plant, we optimize the operating costs of cooling water, vapor and coal for the generation of steam from water, and the investment costs related to the new HEN.

Even though we do not explicitly include environmental impacts in the objective function we are also interested in evaluating the process from an environmental perspective. Specifically, in this work we use the ReCiPe indicator (Goedkoop et al., 2013), available in Ecoinvent Database v.3 (Weidema et al., 2013). This metric is based on the principles of LCA. LCA is a methodology for evaluating the environmental loads associated with a product, process or activity (Guinée et al., 2002). During its application, energy and material used in a process are first identified and qualified. This information is translated into a set of environmental impacts that are aggregated into different groups. These impacts are finally used to evaluate diverse process alternatives that may be implemented in order to achieve environmental improvements. Today, LCA has become the main instrument to evaluate the environmental performance of chemical processes (Azapagic & Clift, 1999; Hoffmann et al., 2001; Petrie & Romagnoli, 2000).

First, raw materials, water, steam and energy consumption per cubic meter of treated water are calculated. Then, land occupied by the plant is estimated in terms of amount of steel used in the plant. This inventory is used to characterize the environmental performance of the process. Regarding the raw materials consumption impact, the sour waters have not been considered as raw materials because they are byproducts from other parts of the petrochemical process. However, the fresh water entering to generate steam is included in the calculation of the impact.

Seventeen categories of impacts are calculated, related to ecosystem quality, human health and resources depletion: agricultural land occupation, climate change (ecosystems), freshwater ecotoxicity, freshwater eutrophication, marine ecotoxicity, natural land transformation, terrestrial acidification, terrestrial ecotoxicity, urban land occupation, climate change (human health), human toxicity, ionizing radiation, ozone depletion, particulate matter formation, photochemical oxidant formation, fossil fuel depletion and metal depletion.

The operating cost of the plant before optimization is $3.3974 million/year and the objective function of the problem without taking into account heat integration is $1.8245 million/year.

shows the values of the independent variables for the plant before optimization and the optimized plant, without heat integration. Even though, this is only an intermediate step it is interesting to compare the environmental impacts of this optimized plant and the base case. If we consider aggregated impacts according to the ReCiPe methodology (Ecosystem Quality, Human Health or Resources Depletion), the optimized plant presents a net reduction in the three indicators (Fig. 5). However, if we consider mid-point indicators, some of them increase with respect to the base case (Fig. 6). Even though this, is beyond the scope of this work, this result shows that other operating conditions could be of interest if some of these environmental indicators must be maintained at lower levels.

< Insert Fig. 5 >

Fig. 5. Comparison of the main impacts between plant before optimization and the optimized plant.

< Insert Fig. 6 >

Fig. 6. Comparison of the mid-point indicators between plant before optimization and the optimized plant.

The optimal solution shows a reduction in steam consumption of 5.501 ton/h. Although total impact decreases with respect to the base case, the indicator “photochemical oxidant formation” increases with respect to the plant before optimization, which is due to the increases in coal requirements. Besides this reduction, it is interesting to note that the flowrate of the recycle stream S1 goes to zero. Consequently, this stream can be removed from the process.

Data of the streams involved in the heat integration including the input and output temperature intervals are shown in Table 6.

Curiously, the optimal solution of this new model for the values of flows and temperatures is the same as the optimal solution of the non-heat integrated model. However, if we run the model removing the purity and recovery constraints –which is equivalent to maximize the heat integration without taking into account the rest of the model-, the total energy consumption is lower than that of the plant before optimization. But, this extra energy saving can be met by increasing the output temperatures of cold streams (S7, S19 and S28) -in the optimized model these temperatures are in their lower bounds-. Increasing some of these temperatures decreases the pollutants recovery in the strippers, as a consequence, the steam flowrate in the stripper must increase to maintain the same recovery, which results in an increase in utility consumption. It is worth remarking that this result is just coincidental and in general the non-heat integrated and heat integrated solutions are different.

Even though the flows and temperatures are the same, utility consumption and environmental impacts are considerably reduced in the heat integrated flowsheet (see Fig. 7).

Table 7 shows optimal values for the main streams in the process. The complete table is too large (93 material streams) and it is included as supplementary material.

The final step consists of generating the HEN. In the literature, different approaches are proposed (a good review can be found in Furman and Sahinidis (2002)). In this paper, we used the superstructure approach presented by Yee and Grossmann (1990). Appendix A shows the equations of the "Pinch Location Method" in a GDP form, which was reformulated as an MINLP model (Grossmann et al., 1998).

Fig. 7 shows the HEN obtained for our case study and Table 8 show the data and the results of the heat integration.

< Insert Fig. 7 >

Fig. 7. Heat exchanger network for the SWS plant.

The operating costs of the heat integrated plant including utilities and the installed cost of the HEN are $0.6486 million/year, which is 80.9 % lower than the base case.

All the models were simulated on Aspen HYSYS v.8.4 in a computer with a 2.60 GHz Pentium® Dual-Core Processor and 4 GB of RAM under

Windows 7. Kriging surrogate models were calibrated using MATLAB (The Mathworks, 2014). As NLP solver, we use CONOPT (Drud, 1996) available through TOMLAB-MATLAB (Holmström, 1999). As MINLP solver, we use a proprietary implementation of a basic Branch and Bound algorithm also interfaced with TOMLAB-MATLAB. The complete model, objective function, explicit constraints, implicit models (models in the process simulator) and surrogate models, are written in a proprietary modeling language (Caballero et al., 2014) interfaced with TOMLAB.

The CPU time used in the optimization of the stripping plant (including sampling, Kriging calibration and model optimization) was around 23 minutes.

Table 9 summarizes the utility needed on the economically optimized plant and on the heat integrated plant.

The described HEN achieves heat recovery and hence lowers heating and cooling requirements (39.0 % and 24.9 %, respectively). This implies a reduction in the cooling water, steam, and coal requirements, with a reduction in the corresponding operating costs and a reduction in the impact by the water, steam and coal supply indexes.

To evaluate the process from an environmental perspective, we use the ReCiPe Endpoint (H,A) indicator (Goedkoop 2013). Table 10 shows the inventory of the processes. All calculations are performed per m^3 of treated water.

In this case, we study the improvement of the process before and after performing the heat exchange network. In Table 11 we can see the final impact obtained in each process. As we can see, the final impact decreases when the heat integration is performed.

Fig. 8 shows the three general categories of impact. Human health and resources depletion are the most affected categories. Impact after heat integration is reduced by about 49.5 % against the impact of the plant before optimization.

< Insert Fig. 8 >

Fig. 8. Comparison of the main impacts for the SWS plant.

Fig. 9 shows the impacts ratios (compared with the base case) of the two alternatives studied. The impacts of each category are normalized by the impacts of the base case ($I_k^n = I_k^i / I_k^{BC}$), where I_k^n is the normalized impact of category k in process i, I_k^i is the categorized impact of category k in process i, and I_k^{BC} is the categorized impact of category k for the base case.

< Insert Fig. 9 >

Fig. 9. Impact ratios of the alternatives studied.

There are some categories of impact, such as terrestrial ecotoxicity, natural land transformation, climate change (ecosystems), particulate matter formation and fossil depletion that have a significant improvement compared with the rest of the categories. After the economic optimization, some categories such as fresh water eutrophication, fresh water ecotoxicity, marine ecotoxicity, ozone depletion and climate change (human health) get worse with respect to the base case, but after the heat integration, all impacts are reduced against the base case.

5. Conclusions

Large scale flowsheet optimization involving «gray box models» has inherent numerical difficulties related to the lack of convergence of some modules, relatively large CPU times for converging some unit operations and the introduction of numerical noise that prevent the accurate estimation of derivatives. To overcome all these difficulties but at the same time maintain as much as possible the rigor and reliability of commercial process simulators, this paper proposes a hybrid approach in which some units are maintained in the process simulator, some units are substituted by surrogate models and we include the possibility of adding explicit constraints in equation form. These constraints can range from simple bounds to complete models in equation form.

We used a disaggregated approach in which we substitute a large surrogate model by a set of smaller surrogate models. An important point consists of identifying which units/modules must be substituted by a surrogate model and which units could be merged in a single surrogate. In the case of chemical process simulators, the most important factors are the CPU time to converge, the numerical noise and the lack of convergence in the complete domain.

In this work we have applied a Kriging interpolation model, with Gaussian extended exponential as spatial correlation functions, to the rigorous optimization of an SWS plant. In this optimization, the stripping columns (implicit black-box functions of the simulator) were substituted by Kriging metamodels. An analysis of degrees of freedom indicated that merging the models of some very integrated columns reduced the complexity of surrogate models and maintained rigor. In this way, surrogate models allow a fast interpolation of new values and have proven to be accurate and reliable.

Economic optimization allows us 46.3 % savings against the plant before optimization, and it allows us to reduce the life cycle assessment of the stripping plant by around 15.1 %.

HENs allows savings in energy (around 39 % in heating and 25 % in cooling) against a plant without heat integration, and they also allow us to reduce the LCA of the plant by around 49.5 %.

Even though the optimization cannot guarantee the global optimum due to the non-convex character of the model. The procedure has proven to be robust, reliable and the final solution obtained is significantly better than the actual one.

Acknowledgement

The authors wish to acknowledge the financial support by the Ministry of Economy and Competitiveness of Spain, under the project CTQ2012-37039-C02-02.

Appendix A. The pinch location method in its disjunctive formulation

The proposed model is based on the same principles as the model of Duran and Grossmann (1986). The basic idea relies on incorporating as a constraint in the process optimization the minimum utility target as a function of the flowrates and temperatures of the process streams. To accomplish this task, they proposed a set of inequalities that rely on a pinch location model and that gives rise to non-differentiabilities which are handled with a smooth approximation. The method consists of determining the pinch candidate with the maximum heat required from heating utilities above the pinch, ensuring that the candidate is a true pinch.

The model proposed by Grossmann et al. (1998) is reformulated as a large MINLP problem, but it can be reduced to an MILP problem when dealing only with isothermal streams, thus guaranteeing the global optimum solution.

A.1. Original simultaneous optimization and heat integration model

The Duran and Grossmann (1986) model for the determination of simultaneous optimization and heat integration is given by the following model (assuming fixed *x*):

$$min: C_S Q_S + C_W Q_W$$

s.t.

$$QSIA(x)^p - QSOA(x)^p \leq Q_S \quad \forall p \in P$$

$$\sum_{i \in H} F_i \left(T_i^{in} - T_i^{out}\right) - \sum_{j \in C} f_j \left(t_j^{out} - t_j^{in}\right) + Q_S - Q_W = 0$$

$$Q_S, Q_W, QSIA, QSOA, F_i, f_j, T_i, t_j \geq 0$$

where

$$QSOA(x)^p = \sum_{i \in H} F_i \left[max\left\{0, T_i^{in} - T^p\right\} - max\left\{0, T_i^{out} - T^p\right\}\right] \tag{A.1}$$

$$QSIA(x)^p = \sum_{j \in C} f_i \left[max\left\{0, f_j^{out} - \left(T^p - \Delta T_{min}\right)\right\} \right.$$
$$\left. - max\left\{0, f_j^{in} - \left(T^p - \Delta T_{min}\right)\right\}\right]$$

$$T^p = T_i^{in} \qquad p = i \in H$$

$$T^p = \left(t_j^{in} + \Delta T_{min}\right) \qquad p = j \in C$$

In this model, *QSOA* is the heat available and *QSIA* is the heat needed above the potential pinch candidate, F_i and f_i are the heat capacity flowrates of hot and cold streams respectively, and *P* stands for the set of pinch candidates of either hot or cold streams.

This method has two main disadvantages. First, the model includes the *max* function in the determination of the heat available for exchange, which is non-differentiable at the value of $T = T^p$. Second, smoothing functions avoid the non-differentiabilities of the max function, but the selection of the parameters can be non-trivial.

Difficulties that are experienced with the Duran and Grossmann (1986) model were overcome with the disjunctive model proposed by Grossmann et al. (1998), who avoided non-differentiabilities and approximations.

A.2. Disjunctive model

The disjunctive model proposed by Grossmann et al. (1998) uses logic disjunctions to explicitly model the relative placement of streams for various potential pinch locations, and explicitly considers the non-isothermal and isothermal streams as separate cases.

Depending on the placement of the streams with respect to pinch temperature, three cases are possible, and only one of them can take place:

1. The hot stream *i* lies completely above the pinch candidate. So all its heat content is available for exchange with the cold streams. This means that both inlet and outlet temperatures are higher than T_k^{in} for a pinch candidate $k \in H$, or higher than $t_k^{in}+\Delta T_{min}$ for candidate $l \in C$. The binary variable YH_i^{Above} means that the hot stream *i* lies above the temperature of hot pinch candidate *k*, and the binary variable YC_i^{Above} means that the hot stream *i* lies above the temperature of the cold pinch candidate plus ΔT_{min}.
2. The hot stream *i* has inlet temperature above the pinch candidate and outlet temperature below it. Only a part of Q_{hot} is available for heat exchange. The binary variables YH_i^{Middle} and YC_i^{Middle} represent the occurrence of this case for hot and cold pinch candidate streams respectively.
3. The hot stream *i* has both inlet and outlet temperatures below the temperature of the pinch candidate, therefore, it cannot exchange heat with the cold streams above the pinch. The binary variables YH_i^{Below} and YC_i^{Below} represent the occurrence of this case for hot and cold pinch candidate streams respectively.

There are three similar options regarding the position of a cold stream *j* with respect to the pinch candidate:

1. The cold stream *j* lies completely above the pinch candidate. So all its heat content is available for exchange with the hot streams. The binary variable ZH_j^{Above} represents the occurrence of stream *j* above the hot pinch candidate stream *k*, and the binary variable ZC_j^{Above} means that the stream *j* lies above the cold stream pinch candidate *l*.
2. The cold stream *j* has outlet temperature above the pinch candidate and inlet temperature below it. Only a part of its heat content is available for heat exchange above the pinch. The binary variables ZH_j^{Middle} and ZC_j^{Middle} represent the occurrence of this case for hot and cold pinch candidate streams respectively.
3. The cold stream *j* has both inlet and outlet temperatures below the temperature of the pinch candidate, therefore, it cannot exchange heat with the hot streams above the pinch. The binary variables ZH_j^{Below} and ZC_j^{Below} represent the occurrence of this case for hot and cold pinch candidate streams respectively.

This situation can be represented in general as a disjunction of three Boolean variables, each one taking a value of true when the constraints that define the case are satisfied, and false otherwise:

$$WH_i^{Above} \underline{\vee} WH_i^{Middle} \underline{\vee} WH_i^{Below} \qquad WC_j^{Above} \underline{\vee} WC_j^{Middle} \underline{\vee} WC_j^{Below}$$

$$WH_i \in \{True, False\}, i = k, l \qquad WC_j \in \{True, False\}, j = k, l$$

The pinch point is located at the inlet temperature of a hot or a cold stream. We explicitly consider both cases:

- The pinch is located at the inlet temperature of a hot stream, or
- The pinch is located at the inlet temperature of a cold stream.

Eq.(A.2) shows the three alternatives for a hot stream.

$$\left[\begin{array}{c} WH_i^{Above} \\ \left[\begin{array}{c} YH_i^{Above} \\ \left.\begin{array}{c} QH_i = F_i\left(T_i^{in} - T_i^{out}\right) = Q_{k,i}^{hp} \\ T_i^{in} \geq T_k^{in} \\ T_i^{out} \geq T_k^{in} \end{array}\right\} k \in H \end{array}\right] \vee \left[\begin{array}{c} YC_i^{Above} \\ \left.\begin{array}{c} QH_i = F_i\left(T_i^{in} - T_i^{out}\right) = Q_{l,i}^{cp} \\ T_i^{in} \geq t_l^{in} + \Delta T_{min} \\ T_i^{out} \geq t_l^{in} + \Delta T_{min} \end{array}\right\} l \in C \end{array}\right] \end{array}\right] \underline{\vee}$$

$$\left[\begin{array}{c} WH_i^{Middle} \\ \left[\begin{array}{c} YH_i^{Middle}\\ \left.\begin{array}{c} QH_i \ge F_i\left(T_i^{in}-T_i^{out}\right) = Q_{k,i}^{hp}\\ T_i^{in}\ge T_k^{in}\\ T_i^{out}\le T_k^{in}\end{array}\right\} k\in H \end{array}\right] \vee \left[\begin{array}{c} YC_i^{Middle}\\ \left.\begin{array}{c} QH_i \ge F_i\left(T_i^{in}-\left[t_i^{in}+\Delta T_{min}\right]\right) = Q_{l,i}^{cp}\\ T_i^{in}\ge t_l^{in}+\Delta T_{min}\\ T_i^{out}\le t_l^{in}+\Delta T_{min}\end{array}\right\} l\in \end{array}\right.\end{array}\right.$$

$$\left[\begin{array}{c} WH_i^{Below} \\ \left[\begin{array}{c} YH_i^{Below}\\ \left.\begin{array}{c} Q_{k,i}^{hp} = 0\\ T_i^{in}\le T_k^{in}\\ T_i^{out}\le T_k^{in}\end{array}\right\} k\in H \end{array}\right] \vee \left[\begin{array}{c} YC_i^{Below}\\ \left.\begin{array}{c} Q_{l,i}^{cp} = 0\\ T_i^{in}\le t_l^{in}+\Delta T_{min}\\ T_i^{out}\le t_l^{in}+\Delta T_{min}\end{array}\right\} l\in C \end{array}\right]\end{array}\right] \forall i\in H$$

Eq.(A.3) shows the three alternatives for a cold stream.

$$\left[\begin{array}{c} WC_j^{Above} \\ \left[\begin{array}{c} ZH_j^{Above}\\ \left.\begin{array}{c} QC_j = f_j\left(t_j^{out}-t_j^{in}\right) = q_{k,j}^{hp}\\ t_j^{in}\ge T_k^{in}-\Delta T_{min}\\ t_j^{out}\ge T_k^{in}-\Delta T_{min}\end{array}\right\} k\in H \end{array}\right] \vee \left[\begin{array}{c} ZC_j^{Above}\\ \left.\begin{array}{c} QC_j = f_j\left(t_j^{out}-t_j^{in}\right) = q_{l,j}^{cp}\\ t_j^{in}\ge t_l^{in}\\ t_j^{out}\ge t_l^{in}\end{array}\right\} l\in C \end{array}\right]\end{array}\right] \underline{\vee}$$

$$\left[\begin{array}{c} WC_j^{Middle} \\ \left[\begin{array}{c} ZH_j^{Middle}\\ \left.\begin{array}{c} QC_j \ge f_j\left(t_j^{out}-\left[T_k^{in}-\Delta T_{min}\right]\right) = q_{k,j}^{hp}\\ t_j^{in}\le T_k^{in}-\Delta T_{min}\\ t_j^{out}\ge T_k^{in}-\Delta T_{min}\end{array}\right\} k\in H \end{array}\right] \vee \left[\begin{array}{c} ZC_j^{Middle}\\ \left.\begin{array}{c} QC_j \ge f_j\left(t_j^{out}-t_l^{in}\right) = q_{l,j}^{cp}\\ t_j^{in}\le t_l^{in}\\ t_j^{out}\ge t_l^{in}\end{array}\right\} l\in C \end{array}\right.\end{array}\right.$$

$$\left[\begin{array}{c} WC_j^{Below} \\ \left[\begin{array}{c} ZH_j^{Below}\\ \left.\begin{array}{c} q_{k,j}^{hp} = 0\\ t_j^{in}\le T_k^{in}-\Delta T_{min}\\ t_j^{out}\le T_k^{in}-\Delta T_{min}\end{array}\right\} k\in H \end{array}\right] \vee \left[\begin{array}{c} ZC_j^{Below}\\ \left.\begin{array}{c} q_{l,j}^{cp} = 0\\ t_j^{in}\le t_l^{in}\\ t_j^{out}\le t_l^{in}\end{array}\right\} l\in C \end{array}\right]\end{array}\right] \forall j\in C$$

The total amount of heating utility Q_S that is required above a pinch candidate is given by an energy balance above the pinch, and the total cooling utility requirement Q_W is given by a general energy balance. The rest of the model can be written as follows:

min: $C_S Q_S + C_W Q_W$

s.t.

$$
\begin{aligned}
& Q_S + \sum_{i \in H} QH_i = Q_W + \sum_{j \in C} QC_j \\
& Q_S \geq \sum_{j \in C} q_{k,j}^{hp} - \sum_{i \in H} Q_{k,i}^{hp} && k \in H \\
& Q_S \geq \sum_{j \in C} q_{l,j}^{cp} - \sum_{i \in H} Q_{l,i}^{cp} && l \in C \\
& QH_i = F_i\left(T_i^{in} - T_i^{out}\right) && i \in H \\
& QC_j = f_j\left(t_j^{out} - t_j^{in}\right) && j \in C
\end{aligned}
\tag{A.4}
$$

Appendix B. Maxmin approach

The 'maxmin' approach distribute a set of points in a domain maximizing the minimum distance between two points.

The maxmin problem can be formulated as an NLP problem as follows:

- Consider the distribution of N points (*i*) in a D-dimensional space (*d*)

$x_{i,d}$ = d dimension component of point i

$$\max \min \left\{distance_{i,j}\right\}$$

$$distance_{i,j} = \sqrt{\sum_{d=1}^{D} \left(x_{i,d}^2 - x_{j,d}^2\right)} \qquad \forall i,j \in Points/i \neq j$$

$$0 \leq x_{i,d} \leq 1$$

In order to avoid maximizing over the result of the min operation that introduces non-differentiabilities, the previous problem can be reformulated transferring the min operation to the constraints and using the auxiliary variable α (the intermediate variable 'distance' can also be removed).

$$\max \alpha$$

$$s.t.\quad \alpha \leq \sqrt{\sum_{d=1}^{D}\left(x_{i,d}^{2}-x_{j,d}^{2}\right)}\qquad \forall i,j\ \in Points/i\neq j$$

$$0\leq x_{i,d}\leq 1$$

Alternatively instead of minimizing the distance, it is possible, without modifying the result, minimize the square of the distance. The only modification proposed in the previous problem consists of fixing 2^D points to the extremes of the interval to avoid extrapolations in the optimization near the 'corners' of the hypercube.

As an example, Fig B.1 shows the distribution of 40 points using the maxmin approach vs a random selection.

< Insert Fig B.1>

Fig B.1. Distribution of 40 points: maxmin approach (left), random selection of points (right).

The idea of fixing 2^D sampling points to the "corners" of the hypercube that defines the domain of the Kriging model is simply to avoid 'extrapolations'. We have numerically checked that results are much more accurate if we avoid any possible extrapolation. For example, Fig B.2 represents 30 points distributed using the maxmin approach in a two-dimensional space. The lines define the boundaries of the convex hull of that set of points. We can only perform interpolations inside the convex hull region. If we fix the points (0,0), (0,1), (1,0) and (1,1), the convex hull includes all the domain.

< Insert Fig B.2>

Fig B.2. Convex hull of 30 points using the maxmin approach in which corners are not explicitly selected as sampled points.

References

Agarwal, A., & Biegler, L. T. (2013). A trust-region framework for constrained optimization using reduced order modeling. Optim Eng, 14, 3-35.

Al-mutairi, E. M. (2010). Optimal design of heat exchanger network in oil refineries. Chem Eng Trans, 21, 955-960.

Allen, B., Savard-Goguen, M., & Gosselin, L. (2009). Optimizing heat exchanger networks with genetic algorithms for designing each heat exchanger including condensers. Appl Therm Eng, 29, 3437-3444.

Aspen Technology, I. (1994-2015). Aspen Technology, Inc. In: Aspen Plus.

Azapagic, A., & Clift, R. (1999). The application of life cycle assessment to process optimisation. Comput Chem Eng, 23, 1509-1526.

Biegler, L. T., Lang, Y. D., & Lin, W. (2014). Multi-scale optimization for process systems engineering. Comput Chem Eng, 60, 17-30.

Caballero, J. A., & Grossmann, I. E. (2008). An algorithm for the use of surrogate models in modular flowsheet optimization. AIChE J, 54, 2633-2650.

Caballero, J. A., Navarro, M. A., Ruiz-Femenia, R., & Grossmann, I. E. (2014). Integration of different models in the design of chemical processes: Application to the design of a power plant. Appl Energy, 124, 256-273.

Conn, A. R., Gould, N. M., & Toint, P. L. (2000). Trust-region methods (MPS-SIAM series on optimization). SIAM, Philadelphia.

Cozad, A., Sahinidis, N. V., & Miller, D. C. (2014). Learning surrogate models for simulation-based optimization. AIChE J, 60, 2211-2227.

Chemstations, I. (2012). Chemstations, Inc. In: CHEMCAD.

Chung, P. S., Jhon, M. S., & Biegler, L. T. (2011). The holistic strategy in multi-scale modeling. In G. B. Marin (Ed.), Advances in Chemical Engineering (Vol. 40, pp. 59-118): Academic Press.

Dantus, M. M., & High, K. A. (1999). Evaluation of waste minimization alternatives under uncertainty: a multiobjective optimization approach. Comput Chem Eng, 23, 1493-1508.

Davis, E., & Ierapetritou, M. (2007). A kriging method for the solution of nonlinear programs with black-box functions. AIChE J, 53, 2001-2012.

Diwekar, U. M. (2003). Introduction to applied optimization. In P. M. Pardalos & D. W. Hearn (Eds.), Applied Optimization. Dordrecht: Kluwer Academic Publishers.

Drud, A. S. (1996). CONOPT: A system for large scale nonlinear optimization. Reference manual. Bagsvaerd, Denmark: ARKI consulting and development A/S.

Duran, M. A., & Grossmann, I. E. (1986). Simultaneous optimization and heat integration of chemical processes. AIChE J, 32, 123.

Eslick, J. C., & Miller, D. C. (2011). A multi-objective analysis for the retrofit of a pulverized coal power plant with a CO2 capture and compression process. Comput Chem Eng, 35, 1488-1500.

Friedman, J. H. (1991). Multivariate adaptative regression splines. Ann Stat, 19, 1-141.

Furman, K. C., & Sahinidis, N. V. (2002). A critical review and annotated bibliography for heat exchanger network synthesis in the 20th Century. Ind Eng Chem Res, 41, 2335-2370.

Gharaie, M., Zhang, N., Jobson, M., Smith, R., & Panjeshahi, M. H. (2013). Simultaneous optimization of CO2 emissions reduction strategies for effective carbon control in the process industries. Chem Eng Res Des, 91, 1483-1498.

Goedkoop, M., Heijungs, R., Huijbregts, M., Schryver, A. D., Struijs, J., & Van Zelm, R. (2013). ReCiPe 2008. A life cycle impact assessment method which comprises harmonised category indicators at the midpoint and the endpoint level.

Grossmann, I. E., Yeomans, H., & Kravanja, Z. (1998). A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration. Comput Chem Eng, 22, A157-A164.

Guinée, J. B., Gorrée, M., Heijungs, R., Huppes, G., Kleijn, R., de Koning, A., van Oers, L., Sleeswijk, A. W., Suh, S., de Haes, H. A. U., de Bruijn, H., van Duin, R., & Huijbregts, M. A. J. (2002). Handbook of Life Cycle Assessment. Operational guide to the ISO Standards. Dordrecht: Kluwer Academic Publishers.

Gutiérrez-Antonio, C., & Briones-Ramírez, A. (2009). Pareto front of ideal Petlyuk sequences using a multiobjective genetic algorithm with constraints. Comput Chem Eng, 33, 454-464.

Gutiérrez-Antonio, C., Briones-Ramírez, A., & Jiménez-Gutiérrez, A. (2011). Optimization of Petlyuk sequences using a multi objective genetic algorithm with constraints. Comput Chem Eng, 35, 236-244.

Hasan, M. M. F., Jayaraman, G., Karimi, I. A., & Alfadala, H. E. (2010). Synthesis of heat exchanger networks with nonisothermal phase changes. AIChE J, 56, 930-945.

Henao, C. A., & Maravelias, C. T. (2011). Surrogate-based superstructure optimization framework. AIChE J, 57, 1216-1232.

Himmelblau, D. M. (2000). Applications of artificial neural networks in chemical engineering. Korean J Chem Eng, 17, 373-392.

Hoffmann, V. H., Hungerbuhler, K., & McRae, G. J. (2001). Multiobjective screening and evaluation of chemical process technologies. Ind Eng Chem Res, 40, 4513-4524.

Holmström, K. (1999). The TOMLAB optimization environment in Matlab. Adv Model Optim, 1, 47-69.

Huang, D., Allen, T. T., Notz, W. I., & Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential kriging meta-models. J Glob Optim, 34, 441-466.

Huang, K. F., & Karimi, I. A. (2013). Simultaneous synthesis approaches for cost-effective heat exchanger networks. Chem Eng Sci, 98, 231-245.

Hyprotech, L. (1995 - 2011). Hyprotech, Ltd. In: HYSYS. Hyprotech Ltd.

Joint Research Centre. (2013). Best available techniques (BAT) reference document for the refining of mineral oil and gas. Institute for Prospective Technological Studies. European IPPC Bureau.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. J Glob Optim, 21, 345-383.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. J Glob Optim, 13, 455-492.

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. Eur J Oper Res, 192, 707-716.

Krige, D. G. (1951). A statistical approach to some mine valuation and allied problems on the Witwatersrand. [Master's thesis]. South Africa: University of Witwatersrand.

Lara, Y., Lisbona, P., Martínez, A., & Romeo, L. M. (2013). Design and analysis of heat exchanger networks for integrated Ca-looping systems. Appl Energy, 111, 690-700.

Leboreiro, J., & Acevedo, J. (2004). Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework. Comput Chem Eng, 28, 1223-1236.

Linnhoff, B., & Flower, J. R. (1978). Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks. AIChE J, 24, 633-642.

Morar, M., & Agachi, P. S. (2010). Review: Important contributions in development and improvement of the heat integration techniques. Comput Chem Eng, 34, 1171-1179.

Navarro-Amorós, M. A., Caballero, J. A., Ruiz-Femenia, R., & Grossmann, I. E. (2013). An alternative disjunctive optimization model for heat integration with variable temperatures. Comput Chem Eng, 56, 12-26.

Palmer, K., & Realff, M. (2002). Metamodeling approach to optimization of steady-state flowsheet simulations. Chem Eng Res Des, 80, 760-772.

Papoulias, S. A., & Grossmann, I. E. (1983). A structural optimization approach in process synthesis. Past II: Heat recovery networks. Comput Chem Eng, 7, 707-721.

Petrie, B. A., & Romagnoli, J. (2000). Process synthesis and optimisation tools for environmental design: methodology and structure. Comput Chem Eng, 24, 1195-1200.

Piela, P. C., Epperly, T. G., Westerberg, K. M., & Westerberg, A. W. (1991). ASCEND: an object-oriented computer environment for modeling and analysis: The modeling language. Comput Chem Eng, 15, 53-72.

Process Systems Enterprise, L. (2000). Process Systems Enterprise, Ltd. In: gPROMS.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., & Tucker, P. K. (2005). Surrogate-based analysis and optimization. Prog Aerospace Sci, 41, 1-28.

Quirante, N., Javaloyes, J., & Caballero, J. A. (2015). Rigorous design of distillation columns using surrogate models based on Kriging interpolation. AIChE J, 61, 2169-2187.

Razib, M. S., Hasan, M. M. F., & Karimi, I. A. (2012). Preliminary synthesis of work exchange networks. Comput Chem Eng, 37, 262-277.

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. J Glob Optim, 56, 1247-1293.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and Analysis of Computer Experiments. Stat Sci, 4, 409-423.

Sasena, M. J. (2002). Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations [Doctor Ph.Thesis]. University of Michigan.

The Mathworks, I. (2014). The Marhworks, Inc. Matlab 8.3. In: Natick, MA: The Mathworks, Inc.

Torres, C. M., Gadalla, M., Mateo-Sanz, J. M., & Jiménez, L. (2013). An automated environmental and economic evaluation methodology for the optimization of a sour water stripping plant. J Clean Prod, 44, 56-68.

Wang, G. G., & Shan, S. (2006). Review of metamodeling techniques in support of engineering design optimization. J Mech Des, 129, 370-380.

Wechsung, A., Aspelund, A., Gundersen, T., & Barton, P. I. (2011). Synthesis of heat exchanger networks at subambient conditions with compression and expansion of process streams. AIChE J, 57, 2090-2108.

Weidema, B. P., Bauer, C., Hischier, R., Mutel, C., Nemecek, T., Reinhard, J., Vadenbo, C. O., & Wernet, G. (2013). Data quality guideline for the ecoinvent database version 3. Overview and methodology. Swiss Centre for Life Cycle Inventories.

Welch, W. J., & Sacks, J. (1991). A system for quality improvement via computer experiments. Comm Stat Theory and Methods, 20, 477-495.

Westerberg, A. W., Hutchison, H. P., Motard, R. L., & Winter, P. (1979). Process flowsheeting (1st ed.). London: Cambridge University Press.

Yee, T. F., & Grossmann, I. E. (1990). Simultaneous optimization models for heat integration - II. Heat exchanger network synthesis. Comput Chem Eng, 14, 1165-1184.

Figure captions:

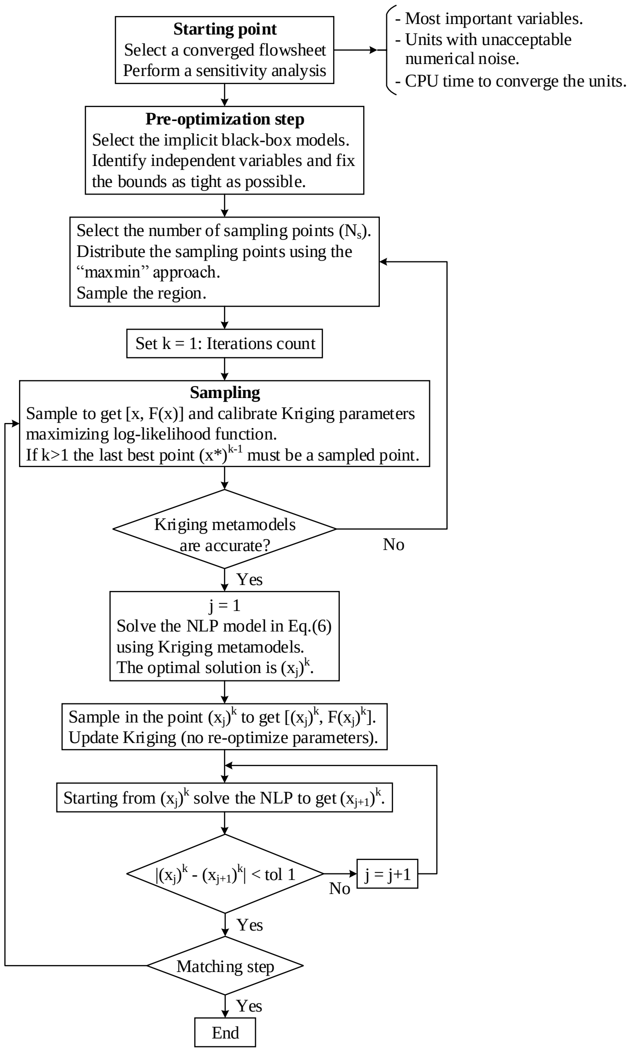

Starting point
Select a converged flowsheet
Perform a sensitivity analysis
- Most important variables.
- Units with unacceptable numerical noise.
- CPU time to converge the units.
Pre-optimization step
Select the implicit black-box models.
Identify independent variables and fix the bounds as tight as possible.
Select the number of sampling points (N_s).
Distribute the sampling points using the "maxmin" approach.
Sample the region.
Set k = 1: Iterations count
Sampling
Sample to get [x, F(x)] and calibrate Kriging parameters maximizing log-likelihood function.
If k>1 the last best point $(x^*)^{k-1}$ must be a sampled point.
Kriging metamodels are accurate?
No
Yes
j = 1
Solve the NLP model in Eq.(6) using Kriging metamodels.
The optimal solution is $(x_j)^k$.
Sample in the point $(x_j)^k$ to get $[(x_j)^k, F(x_j)^k]$.
Update Kriging (no re-optimize parameters).
Starting from $(x_j)^k$ solve the NLP to get $(x_{j+1})^k$.
$|(x_j)^k - (x_{j+1})^k| <$ tol 1
No
j = j+1
Yes
Matching step
Yes
End

Fig. 1. Flowchart of Kriging based NLP optimization algorithm.

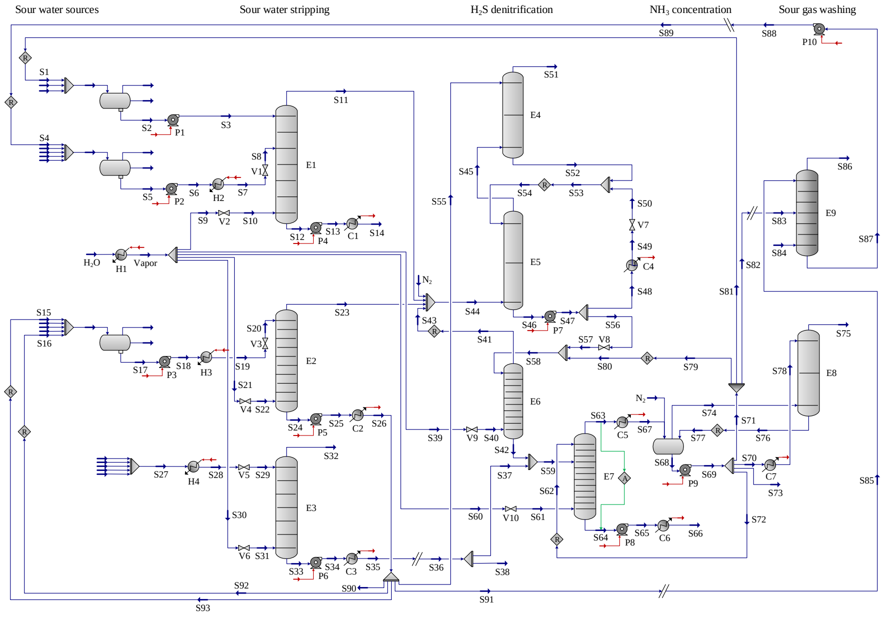


Fig. 2. Complete flowsheet of the sour water stripping plant. The flowsheet has been obtained from a work by Torres et al. (2013) where the heat exchanger network has been removed.

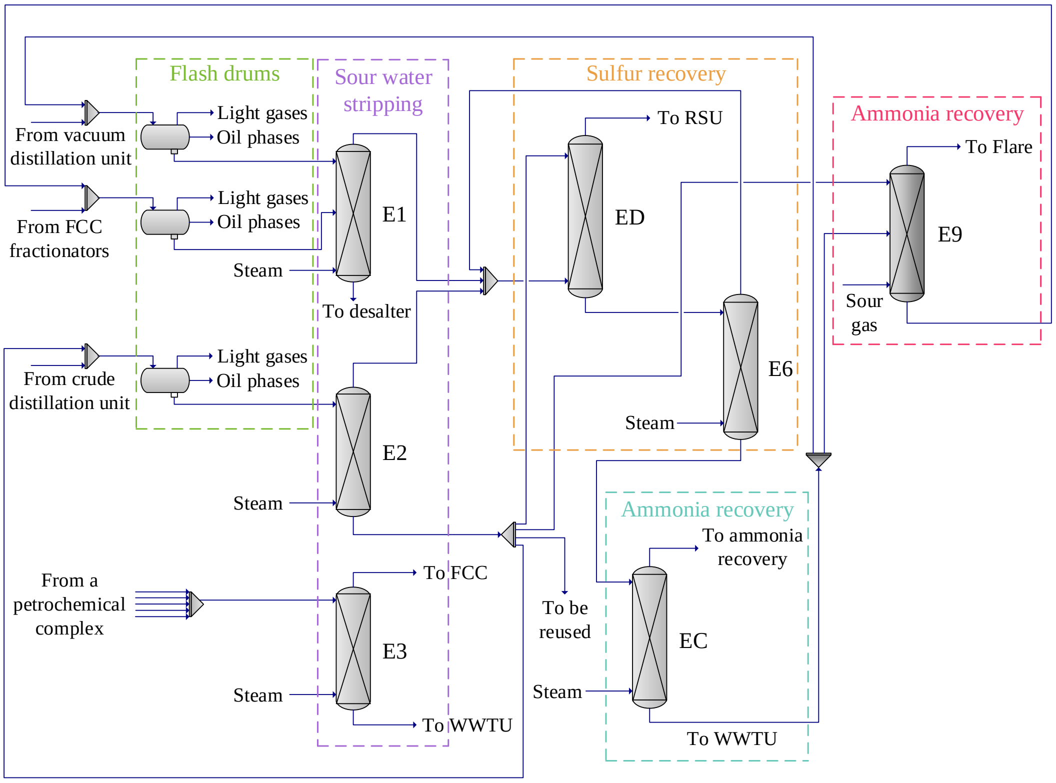


Fig. 3. Simplified process flowsheet of the SWS plant. Note that unit ED is the result of merging the original columns E4 and E5 in a single surrogate model, and unit EC is the result of merging the original columns E7 and E8 in a single surrogate model. For the sake of clarity, in the scheme, pumps, valves, coolers and heaters have been removed.

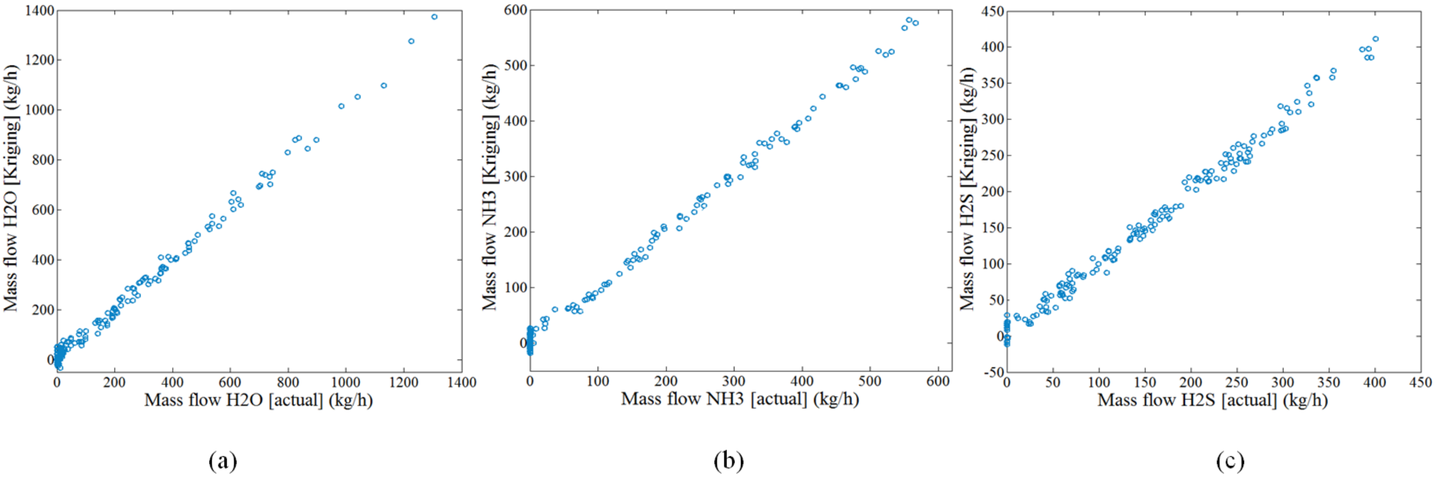


Fig. 4. Cross-validation for vapor obtained from unit ED. (a) mass flow H_2S, (b) mass flow NH_3, and (c) mass flow H_2O.

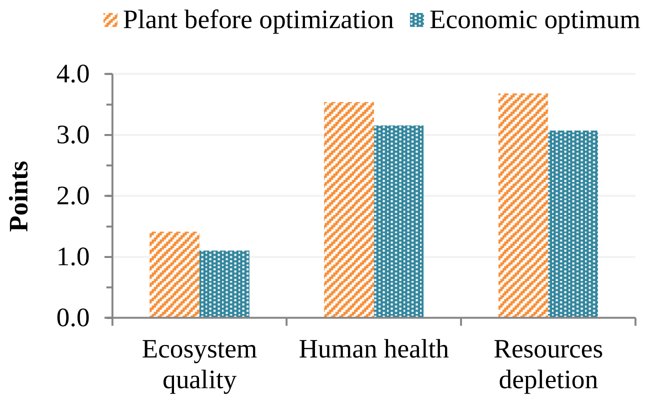


Fig. 5. Comparison of the main impacts between plant before optimization and optimized plant.

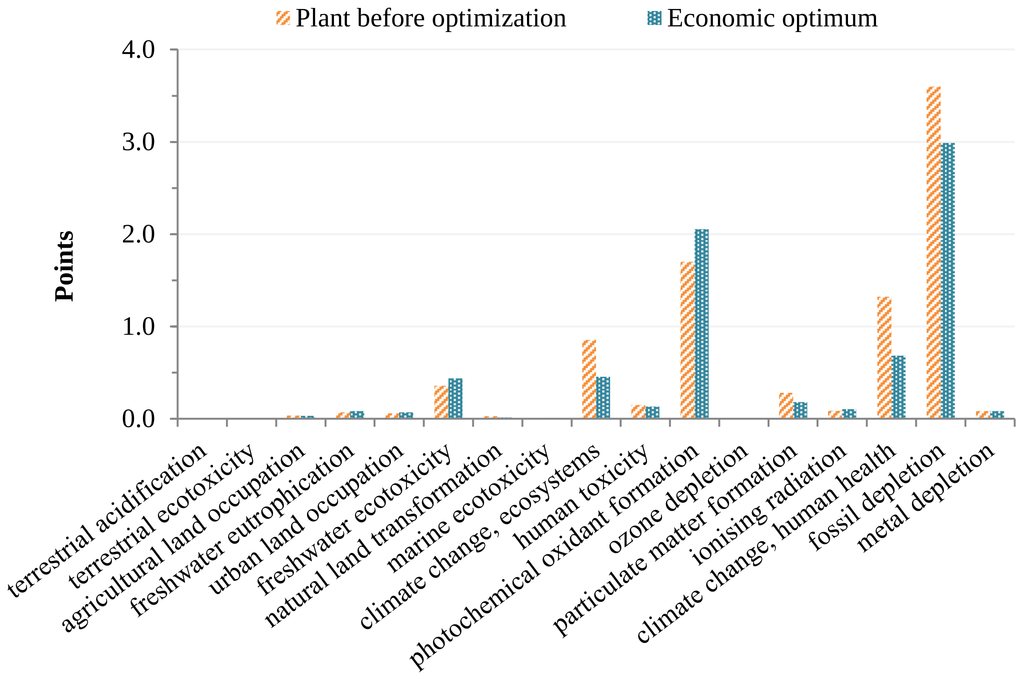


Fig. 6. Comparison of the mid-point indicators between plant before optimization and optimized plant.

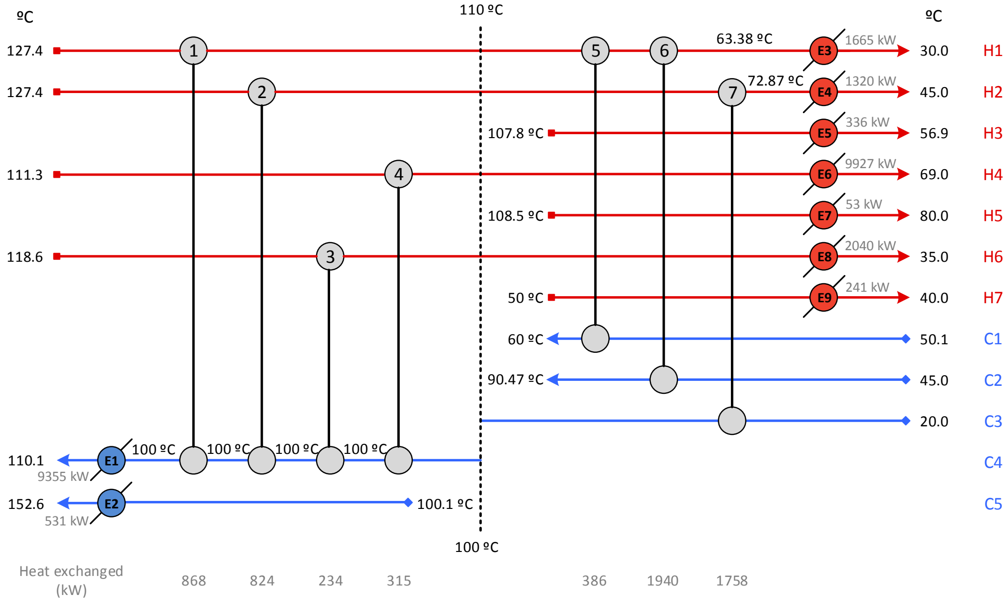


Fig. 7. Heat exchanger network for the SWS plant.

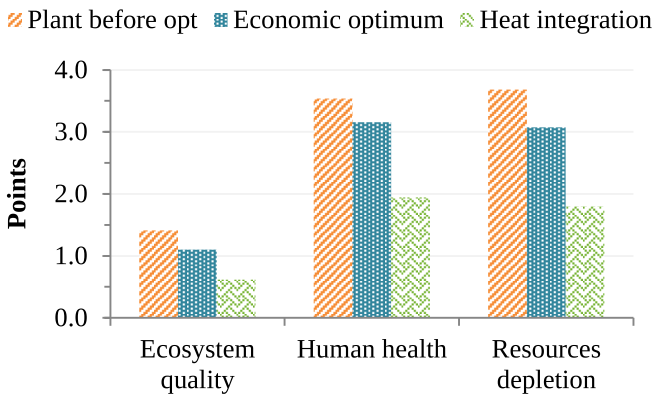


Fig. 8. Comparison of the main impacts for the SWS plant.

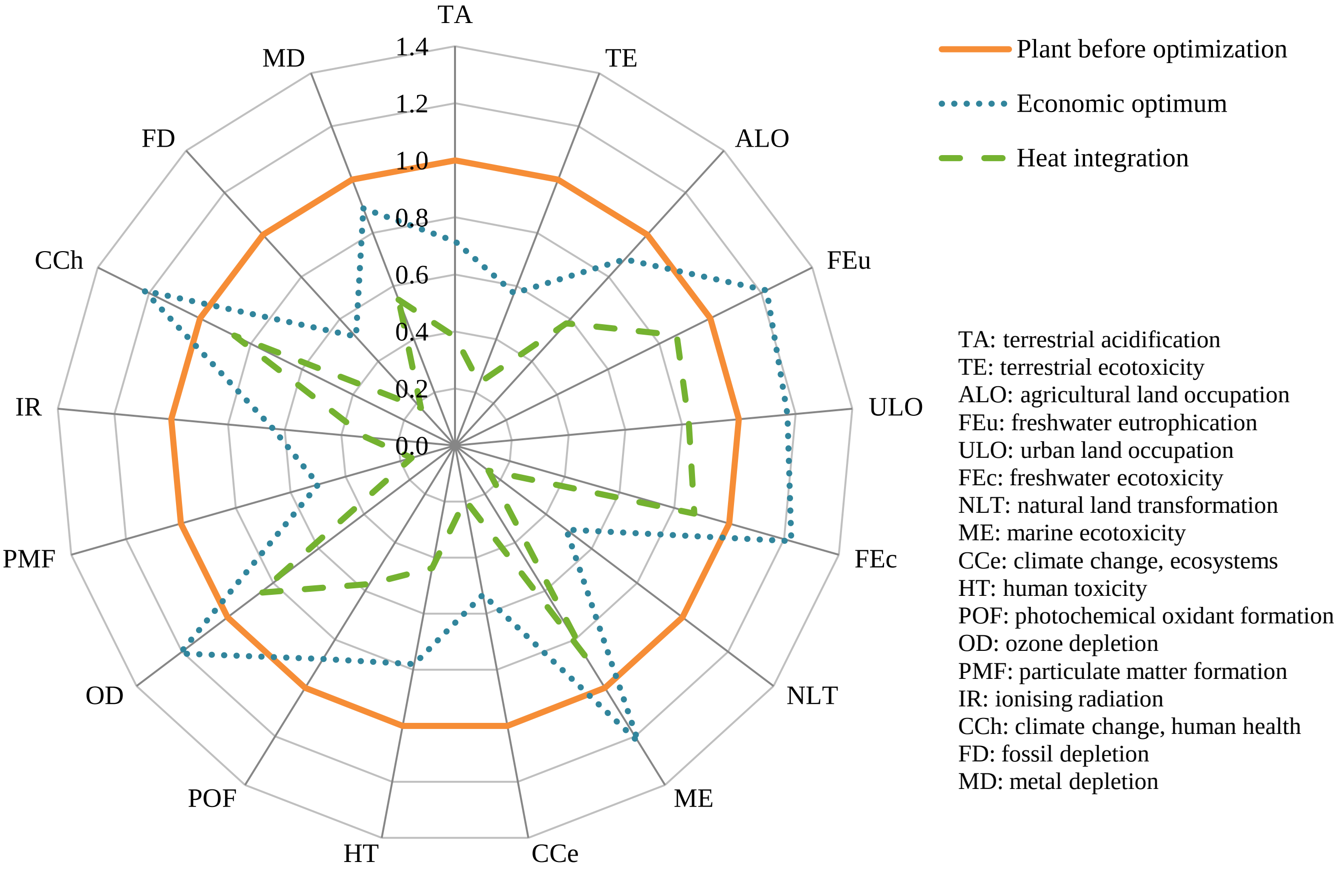


Fig. 9. Impact ratios of the alternatives studied.

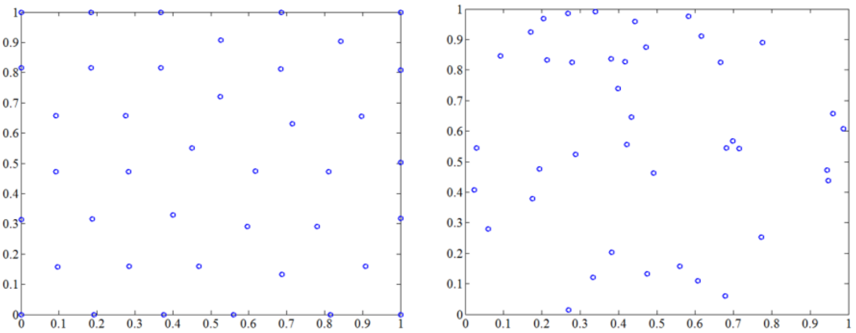


Fig. B.1. Distribution of 40 points: maxmin approach (left), random selection of points (right).

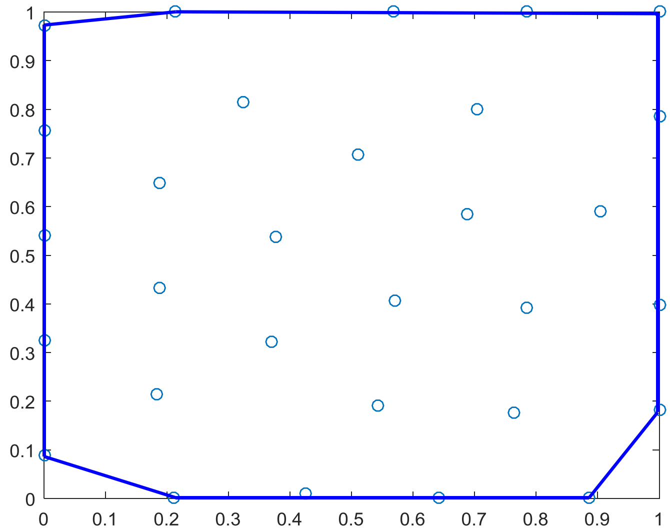


Fig. B.2. Convex hull of 30 points using the maxmin approach in which corners are not explicitly selected as sampled points.

Table 1. Relevant data to streams and equipment.

Streams									
	From vacuum distillation unit			From FCC fractionators			From crude distillation unit		
Mass flow (kg/h)									
H_2S	23.00			26.57			932.42		
NH_3	0.00			0.00			0.00		
H_2O	3,108.00			24,732.24			10,895.10		
N_2	1.00			0.15			46.37		
CH_4	80.00			12.08			3,709.32		
n-Decane	1.00			7.84			1.70		
Temperature (ºC)	44.70			52.27			44.13		
Pressure (kPa)	110.00			100			400.00		
	Wash water of ethanol and ETBE streams								
Mass flow (kg/h)									
H_2S	0.00								
NH_3	0.00								
H_2O	4,500.00								
ETBE	1.00								
ethanol	1,000.00								
Temperature (ºC)	34.73								
Pressure (kPa)	400.00								
Equipment									
	E1	E2	E3	E4	E5	E6	E7	E8	E9
Trays	7	10	7	3	3	10	12	3	7
Feed tray	3	-	-	-	-	-	4	-	4
$P_{overhead}$ (kPa)	200	200	120	150	160	200	165	150	100
$P_{bottoms}$ (kPa)	250	250	150	160	200	250	190	160	100

Table 2. Specifications for the outputs streams.

Specification	**Value**	
Water to desalter	$[NH_3] \leq 50$ ppm	$[H_2S] \leq 10$ ppm
Water to be reused	$[NH_3] \leq 50$ ppm	$[H_2S] \leq 10$ ppm
Water to WWTU	$[NH_3] \leq 50$ ppm	$[H_2S] \leq 10$ ppm
Ethanol recovery	≥ 80%	
H_2S removed	≥ 85%	
Gas to the flare	$[H_2S] \leq 15$ ppm	
Ammonia-rich gas	NH_3 composition ≥ 79% w/w	

Table 3. Input-output Kriging models used.

	Inputs	**Outputs**
E1	Temperature (S7) Mass flow steam (S10)	Recovery (H_2S, NH_3, H_2O) (S11) Temperature (S11, S13), Diameter
E2	Temperature (S19) Mass flow steam (S22)	Mass flow (H_2S, NH_3, H_2O) (S23) Temperature (S23, S25), Diameter
E3	Temperature (S28) Mass flow steam (S31)	Mass flow (Ethanol, H_2O) (S32) Temperature (S34), Diameter
ED	Mass flow H_2O (S55) Mass flow (H_2S, NH_3, H_2O) (S44)	Mass flow (H_2S, NH_3, H_2O) (S51) Temperature (S57), Diameter
E6	Mass flow (H_2S, NH_3, H_2O) (S58) Mass flow steam (S40)	Mass flow (H_2S, NH_3, H_2O) (S41) Temperature (S41, S42), Diameter
EC	Mass flow (H_2S, NH_3, H_2O) (S59) Mass flow steam (S61)	Mass flow (H_2S, NH_3, H_2O) (S66, S71, S75) Temperature (S65, S71), Diameter E7, Diameter E8
E9	Mass flow H_2O (S85) Mass flow (H_2S, NH_3, H_2O) (S83)	Mass flow (H_2S, NH_3, H_2O) (S86) Temperature (S87), Diameter

.

Table 4. Parameters of the Kriging models

	μ	σ^2	θ				CPU time (s)
recovery H_2S (S11)	0.9905	$1.8322\cdot10^{-4}$	51.2398		102.4619		0.3906
recovery NH_3 (S11)	0.9409	0.0052	49.8325		98.8359		0.2465
recovery H_2O (S11)	0.1259	$4.7678\cdot10^{-4}$	4.5785		6.4519		0.2351
Temperature (S11)	119.4329	0.2864	64.9716		78.2388		0.2410
Temperature (S13)	127.4196	0.0017	48.6678		98.7932		0.2480
Diameter E1	1.0058	0.0019	21.2679		49.8229		0.2793
H_2S (S23)	54.8384	0.3371	30.3657		44.4215		1.2396
NH_3 (S23)	23.5720	8.0419	31.1464		39.8133		0.9832
H_2O (S23)	$3.2565\cdot10^{3}$	$1.1210\cdot10^{6}$	172.3468		22.2216		1.0001
Temperature (S23)	127.4165	$7.5451\cdot10^{-4}$	30.3368		40.3506		1.0363
Temperature (S25)	119.5658	0.0494	68.0711		37.2476		1.9473
Diameter E2	0.8781	0.0019	191.4413		26.7465		1.2153
Ethanol (S32)	479.5459	$6.0242\cdot10^{3}$	13.0197		8.2794		1.4458
H_2O (S32)	788.7408	$6.2387\cdot10^{3}$	14.5060		12.9042		1.1096
Temperature (S34)	108.1987	1.7976	29.2882		30.8195		1.3372
Diameter E3	0.4664	$8.7215\cdot10^{-5}$	34.9869		8.4743		1.4112
H_2S (S51)	172.0015	$3.5525\cdot10^{3}$	1.8217	8.0369	7.9197	3.7908	14.5655
NH_3 (S51)	142.8340	$5.2646\cdot10^{3}$	3.5769	1.0978	15.2870	2.8074	24.2176
H_2O (S51)	223.4028	$2.0460\cdot10^{4}$	6.0966	0.7495	14.3038	3.7687	21.1290
Temperature (S57)	104.6240	19.2317	1.1404	0.1541	2.6022	113.1199	20.2154
Diameter ED	1.6540	0.2527	96.8494	168.5284	247.2531	91.7464	4.2359
H_2S (S41)	186.2353	$1.4950\cdot10^{3}$	7.3879	5.8182	1.5250	5.5275	13.8337
NH_3 (S41)	295.5981	$7.1410\cdot10^{3}$	7.1511	0.0661	7.2842	11.5277	15.5380
H_2O (S41)	$9.2640\cdot10^{2}$	$4.8720\cdot10^{4}$	8.2278	0.6201	1.9402	9.3895	12.6332
Temperature (S41)	108.8436	3.3279	8.7663	0.9607	6.0040	10.5145	20.8880
Temperature (S42)	123.7736	3.4871	21.0310	0.2882	3.8454	12.7050	24.1590
Diameter E6	0.5167	0.0039	175.8764	98.8710	143.0506	196.3369	6.2031
H_2S (S75)	23.2083	829.9996	7.5695	227.3983	1.3714	2.5050	17.1013
NH_3 (S75)	31.4673	5414.4000	135.2778	166.3338	170.3435	140.7423	4.2125
H_2O (S75)	3.0036	23.8063	141.3607	159.5560	169.8580	141.2453	8.0804
H_2S (S66)	0.2952	1.5235	127.8927	76.4676	205.2807	197.0875	4.8603
NH_3 (S66)	1.5943	13.5972	0.3761	2.6559	34.0324	26.1304	12.8347
H_2O (S66)	$1.3601\cdot10^{4}$	$3.9348\cdot10^{6}$	0.7795	1.2578	16.4600	0.8558	9.1953

Temperature (S65)	118.5636	0.0039	0.4320	2.6319	25.3956	26.7801	10.8941
H_2S (S71)	43.8709	$1.6997\cdot10^{3}$	144.3157	136.7316	159.2865	138.6298	7.0768
NH_3 (S71)	100.8917	$3.7155\cdot10^{3}$	133.0920	180.2039	124.6850	153.4176	5.3456
H_2O (S71)	655.2293	$7.2195\cdot10^{3}$	136.8362	152.7180	145.4301	138.9548	5.9878
Temperature (S71)	51.5451	18.8377	127.8658	176.2773	141.4100	164.1774	7.9740
Diameter E7	0.7450	0.0071	145.4363	141.8550	129.9919	168.3668	5.5577
Diameter E8	1.0082	0.0599	151.4114	164.1734	158.8804	161.4596	5.6695
H_2S (S86)	18.4525	41.0718	0.0559	13.5375	15.1862	0.7849	12.4399
NH_3 (S86)	0.2665	0.0714	9.3878	14.8241	18.1175	0.2389	10.6068
H_2O (S86)	65.6713	0.9515	4.9754	5.5652	14.5044	3.1562	10.8457
Temperature (S87)	36.7473	0.2797	14.1687	2.8448	3.5022	3.4658	9.9497
Diameter E9	0.4716	$3.9732\cdot10^{-5}$	17.4777	1.9550	4.6635	1.0851	6.4787

Table 5. Values of independent variables for the plant before optimization and the optimized plant.

	Temperature (ºC)				**Mass flow (kg/h)**					**C_{op} ($MM/year)**
	S7	S19	S28		S10	S22	S31	S40	S61	
Plant before optimization	132.20	115.00	79.98		6,162.00	4,548.00	521.00	1,000.00	2,800.00	3.3974
Optimized plant	60.00	90.47	35.00		7,525.14	4,664.74	1,331.58	1,533.00	3,287.30	1.8245

Table 6. Data of streams involved in the heat integration.

	T_{in} (ºC)	T_{out} (ºC)	$F \cdot C_p$ (kW/ºC)	Type
H1	127.40	30.00	49.8952	Hot
H2	127.40	45.00	47.3606	Hot
H3	107.80	56.91	6.6028	Hot
H4	111.30	69.00	242.1299	Hot
H5	108.50	80.00	1.8687	Hot
H6	118.60	35.00	27.1970	Hot
H7	50.03	40.00	24.1778	Hot
C1	50.06	60.00	38.8429	Cold
C2	45.00	90.47	42.6701	Cold
C3	34.73	35.00	6.2115	Cold
C4	20.00	100.00	21.9798	Cold
C5	100.00	100.10	$1.1596 \cdot 10^5$	Cold
C6	100.10	152.60	10.1238	Cold

Table 7. Summary of the optimal values for the main streams of the SWS plant.

	Temperature (ºC)				**Mass flow (kg/h)**					**C_{op} (\$MM/year)**
	S7	S19	S28		S10	S22	S31	S40	S61	
Optimized plant	60.00	90.47	35.00		7,525.14	4,664.74	1,331.58	1,533.00	3,287.30	0.6486

Table 8. Summary of the results obtained through HEN model.

Q_{heat}	9,886.806 kW	
Q_{cool}	15,582.425 kW	
C_{op}	0.6486 $MM/year	
Pinch point	100-110 ºC	
ΔT_{min}	10 ºC	
Exchanger	**Heat (kW)**	**Exchange area (m^2)**
1	868	100.771
2	824	95.652
3	315	59.198
4	234	33.767
5	386	27.957
6	1940	204.850
7	1758	137.803
E1	9355	314.786
E2	531	43.644
E3	1665	211.572
E4	1320	91.809
E5	336	13.232
E6	9927	337.229
E7	53	1.660
E8	2040	115.015
E9	243	33.584

Table 9. Summary of the minimum utility needed.

	Q_{heat} (kW)	**Q_{cool} (kW)**	**C_{op} ($MM/year)**
Economically optimized plant	16,212.640	20,755.100	1.8245
Heat integrated plant	9,886.806	15,582.425	0.6486

Table 10. Inventory of the different alternatives.

	Units	**Plant before optimization**	**Economic optimum**	**Heat integrated**
steel	kg	0.0022	0.0022	0.0022
electricity	kWh	0.8139	1.0277	1.0277
steam	MJ	341.5092	121.2770	0.0000
tap water	kg	22,665.0800	26,106.0600	19,655.8800
coal	kg	18.7777	23.0341	16.3991

Table 11. Final impact of the different alternatives.

	Impact (points/m^3 treated water)
Plant before optimization	8.6342
Economically optimized plant	7.3308
Heat integrated plant	4.3587