

Network Synthesis Including Detailed Packed Bed Column Designs in Python MExNetS - An Open-Source Package for Mass Exchanger

Michael Short^{a,*}, Adeniyi J. Isafiade^b

^aDepartment of Chemical and Process Engineering, University of Surrey, Guildford, UK

^bDepartment of Chemical Engineering, University of Cape Town, South Africa

m.short@surrey.ac.uk

Mass exchanger networks (MENs) are used to remove/recover contaminants from polluted streams through absorption with available process streams or external mass separating agents. Process Integration techniques such as Pinch Technology (PT) or mathematical optimisation can be used to synthesise optimal networks, however a lack of accessible software and difficulties in formulating the non-convex problem has stunted research. This article presents an open-source Python package for the synthesis of optimal MENs. The package uses the algebraic modelling language, Pyomo, and takes advantage of Python's object-oriented nature to solve a series of optimisation problems, improving on the performance of previous approaches to the problem of incorporating detailed unit designs into MEN synthesis. The package uses automated initialisation strategies to first solve a superstructure-based mixed-integer nonlinear program (MINLP). Thereafter, a detailed optimisation model, formulating the packed column as a system of differential-algebraic equations, is used to design the columns. This detailed packed column design is used to update the MINLP through correction factors, driving the network solution towards the detailed unit optimisation solutions. The new software, called MExNetS, implements this strategy in a user-friendly package that is easily modified and well-documented. In addition to the new software implementation, novel strategies are employed to ensure feasibility at each iteration, which is a challenge in these non-convex optimisation formulations, and new binary cuts are generated and applied to the MINLP that can significantly speed up convergence compared to the previous study. The package also contains automatic superstructure generation based on user-inputted data, with the hope that this software can inspire further research in this area and be accessible to practitioners.

1. Introduction

One of the key modern challenges in the chemical and process industries is how to reduce harmful emissions into the environment. One way in which to do this is using Mass Integration (MI), where process streams or externally sourced mass separating agents (MSAs) are used to absorb pollutants present in waste streams before being discharged to the environment. MENs can be synthesised to optimally match potential MSAs or other process (lean) streams with polluted (rich) streams to design the network that has the lowest total annual cost (TAC) based on the purchase of external MSAs and column capital costs. Approaches to MEN synthesis (MENS) have typically been divided into sequential or simultaneous optimisation methods. In sequential approaches, thermodynamic targets for minimum external MSA usage and the minimum number of units are set *a priori* and then a network is designed to get as close as possible to these targets (El-Halwagi and Manousiouthakis, 1989). Oladosu et al. (2020) have recently extended these concepts by presenting a new algebraic tool, able to simultaneously target and design MENs. These approaches cannot account for multiple targets such as capital costing and operating costs simultaneously. In simultaneous methods, the system is represented as a mathematical program, which can then be solved using a mixed-integer nonlinear programming (MINLP) solver. This introduction will briefly summarise simultaneous approaches to MENS with a focus on detailed unit modelling aspects. For a more detailed review of MENS technology, readers are referred to Isafiade and Short (2019).

Papalexandri et al. (1994) first introduced simultaneous MINLP formulations for MENS through the introduction of a hyperstructure-based method. The formulation was highly non-convex and excluded certain network configurations. Hallale and Fraser (2000), using PT for MI, were the first to include capital cost targets and detailed column models into the network synthesis, however the approach cannot account for capital and MSA costs simultaneously. Other approaches to MENS have used superstructure methods based on the stagewise superstructure (SWS) approach of Yee and Grossmann (1990), originally used for heat exchanger network synthesis (HENS). The first to adopt this formulation to MENS was Chen and Hung (2005) with a highly non-convex MINLP, dealing with multiple contaminants and staged columns. Szitkai et al. (2006) presented an MINLP formulation that used mostly linear constraints by assuming iso-compositional mixing, packed columns and single contaminants, with a non-convex formulation also presented for multiple contaminants. Isafiade and Fraser (2008) presented a different superstructure formulation that used compositions of supply and target compositions of streams to define superstructure boundaries, which was later extended by Azeez et al. (2012) to define interval boundaries by supply compositions only. These different restructurings of the superstructure allow for more efficient problem boundaries and initialisations, while also decreasing binary variables, which tend to make the problem more difficult to solve.

Noting that network synthesis techniques assumed that column diameters were fixed, Isafiade and Short (2016) directly embedded more detailed aspects of column design in the MINLP. This model included diameter as an optimisation variable, resulting in additional variables for fluid velocities and mass transfer coefficients, with column flooding also considered. Including these details directly in the MINLP results in an increasingly non-convex model formulation that requires multi-starting procedures to obtain initialisations that result in reasonable feasible solutions and result in low chances of finding global solutions.

Short et al. (2018) considered the impact of including detailed column models into the network optimisation. Their work used correction factors in the MINLP to avoid increasing non-convexity of the formulation to include detailed model information, based on an algorithm developed for HENS (Short et al., 2016). The technique ensures that both detailed unit models and the network were locally optimal, however long solution times and high non-convexity made automation challenging, with repeated starts common due to poor initialisation strategies. This method is described in more detail in Section 2, as it forms the basis of the current study.

In terms of software, there are few tools available for MI, and fewer still that incorporate simultaneous mathematical optimisation and open-source implementation. El-Halwagi's (1997) book included a PT-based MI tool called MEN, written in the LINGO optimisation language. There are several packages for MI via water networks and hydrogen networks available from The University of Manchester's Centre for Process Integration, but these are not available freely, are based on PT, and do not include both network and detailed packed column simultaneous optimisation. Ng et al. (2014) released RCNet for the optimisation of resource conservation networks, which deals with several recovery problems, however it is implemented within Microsoft Excel and is unsuitable to nonlinear optimisation.

To address the lack of open-source and non-PT-based MENS tools online, this paper introduces a software package, made freely available online, that implements an enhanced and modified version of the algorithm from Short et al. (2018). The package, called MExNetS, allows users to input their own data, use a variety of solvers, and test and experiment with settings and individual unit models. The package enhances the previous work by allowing for a variety of superstructure types to be chosen by the user and automatically implemented, as well as having an automated, rigorous initialisation procedure that removes manual user interference to ensure that the highly nonlinear models find feasible solutions at each iteration. The modular nature of an object-oriented programming language such as Python, also means that those with research interests in only the detailed NLP packed column exchanger optimisation models or the network optimisation MINLP model can easily access and modify these aspects.

2. Methodology

The new software package uses most of the algorithmic ideas and model formulations from the study of Short et al. (2018). In this method, an MINLP formulation is first solved with a set of initial fixed parameters for mass transfer coefficients, column diameters, packing costs, etc. This problem can be challenging to solve as the log mean composition difference (LMCD), as well as the costing functions, are all nonlinear. In the original method, much time was spent initialising and bounding the model and choosing appropriate MINLP solver options to obtain reasonable or feasible solutions. The new software approach presented here, automatically does this for the user by first solving an NLP problem, with all binary variables fixed to obtain a feasible solution to start the MINLP solver. The MINLP model formulation here is based on either the SBS (supply-based superstructure) or SWS superstructure approaches (the user decides, and the superstructure is automatically constructed). The model is the same as used by Short et al. (2018), which itself is an adaptation from Azeez et al. (2012), with additional correction factors and costing information relating to column internals. Correction factors are included

to ensure that solutions in the MINLP converge to the same results obtained in the detailed unit models. The objective function is shown below in Eq. 1 and all other model equations can be found in the original study.

$$\min \left\{ AF \left\{ \sum_{r \in R} \sum_{l \in S} \sum_{k \in K} \left(\left[CC \cdot (Dcor_{r,l,k} \cdot D_{r,l,k})^{CE} \cdot 1.15 \cdot Hcor_{r,l,k} \cdot H_{r,l,k} \right] + \left[PackCost_{r,l,k} \cdot \frac{\pi}{4} \cdot (Dcor_{r,l,k} \cdot D_{r,l,k})^2 \cdot Hcor_{r,l,k} \cdot H_{r,l,k} \right] \right) \right\} + FC \left(\sum_{r \in R} \sum_{l \in S} \sum_{k \in K} (z_{r,l,k}) \right) + \left[\sum_{l \in S} AC_l \cdot L_l \right] \right\} \quad (1)$$

In this equation AF is an annualisation factor, $D_{r,l,k}$ is diameter of packed column associated with rich stream r , lean stream l , in interval k , with associated correction parameter $Dcor_{r,l,k}$. $H_{r,l,k}$ is packed column height, with associated correction, $Hcor_{r,l,k}$. The $PackCost_{r,l,k}$ is the cost of the packing within the mass exchanger, which is corrected at each iteration based on the packing size. $z_{r,l,k}$ is a binary variable, with FC as the fixed cost of a column. AC_l is the cost of the MSAs and L_l is the MSA flowrate. A list of all correction factors used in this study are presented in Table 1. CC and CE are problem specific cost factors.

Table 1: The code structure used within MExNetS

Correction Factor	Purpose
$kyCor_{r,l,k}$	Correction for the overall mass transfer coefficient, $ky_{r,l,k}$
$aiCor_{r,l,k}$	Correction for the interfacial area of the packing, $ai_{r,l,k}$
$Hcor_{r,l,k}$	Correction to column height, $H_{r,l,k}$
$Dcor_{r,l,k}$	Correction for column diameter, $D_{r,l,k}$
$PackCost_{r,l,k}$	The updated packing cost obtained from the NLP step

During solution of the MINLP, several solvers can potentially be called, should the current MINLP solver fail to obtain a feasible solution with reasonable optimality gap. Using the GAMS (GAMS Development Corporation) solver interface with Pyomo, several solvers can be used, with checks as to whether the optimal solution is found with each solve. Usually, an optimal MINLP solution is found within 2 CPUs to 300 CPUs depending on the size of the problem and the number of solvers used. It can be argued here that for this specific algorithm, which uses a multi-start procedure, that feasibility and reasonably good solutions are more important than provable global optimality. A further improvement upon the original study, implemented within MExNetS, is that after the MINLP is solved, a further NLP sub-optimisation is run, in which the iso-compositional mixing assumption is relaxed. In this NLP, binary variables are fixed, and additional mixing constraints are added to allow for split streams to mix at different compositions.

Once a solution to the MINLP is obtained, the resulting mass balances, and inlet and outlet conditions to the columns, are taken as boundary conditions to the detailed design optimisation models. These models include variables for mass transfer coefficients, column diameters, packing sizes, and flooding equations. The differential equations in the system are discretised along the height of the column via orthogonal collocation on finite elements (FEs) using Radau polynomials and the entire system of columns is solved simultaneously. Note that in the MINLP, linear phase equilibrium relations are assumed, but in the detailed NLP models any smooth phase equilibrium relationship could be used, and this method could be used in many diverse separation applications. In the original paper from Short et al. (2018) all columns within the network were solved simultaneously with 200 FEs. To solve this large NLP problem (upwards of 30,000 variables), a series of initialisation stages were required that often required manual user intervention. Within MExNetS, the columns are solved separately, allowing for a smaller number of FEs and no specialised initialisation scheme for most columns. When a column model fails to find to a feasible solution, the algorithm automatically increases the discretisation and calls upon an initialisation routine whereby progressively more complex problems with increasing numbers of variables are solved. This increases the robustness of the algorithm and reduces the need for manual interference from users.

Once the optimal detailed column designs are obtained for the network, a set of correction factors are calculated to update the parameters (in Table 1) in the MINLP model, forcing the MINLP towards the solution of the detailed exchangers. The MINLP model is then re-run to obtain a new network based on the updated values. This is run until convergence is reached (NLP and MINLP objective functions are within tolerance), or until a maximum number of iterations is reached. The power of this method is that the non-convex MINLP is multi-started with perturbed initial conditions repeatedly, increasing the chances of finding a globally optimal solution. The detailed column network is solved to an optimal point at each iteration, ensuring that the results are validated, and

unrealistic solutions resulting from the shortcut models in the MINLP are updated towards feasible networks. The algorithmic details and code structure of the package are shown in Figure 1.

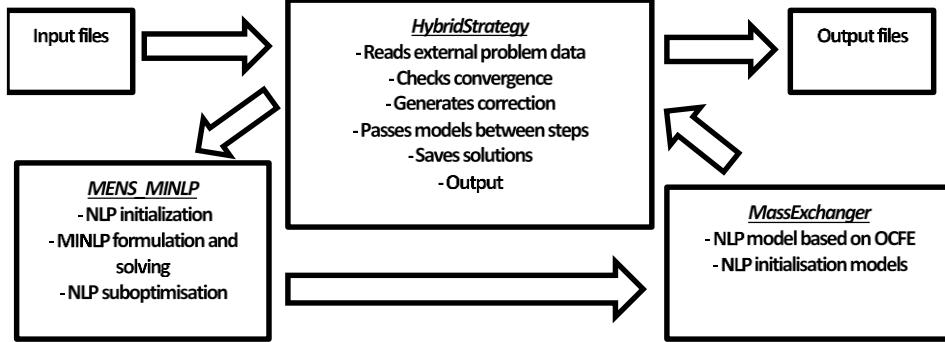


Figure 1: The code structure used within MExNetS

The algorithm is as follows:

1. User input and problem set-up. Read data and input user-selected options such as whether to store binary cuts or not, which superstructure to select, tolerances, etc. Correction factors are set to 1 and parameters take initial values. Set iteration counter, $k = 1$, allow all binary combinations to be selected.
2. *MENS_MINLP*: In this section, we build all the necessary constraints for the MINLP model formulation with iso-compositional mixing, selected superstructure, and current values for correction factors.
 - a. First solve an initialisation NLP that fixes all binary variables to 1.
 - b. Attempt to solve with BARON (Tawarmalani and Sahinidis, 2005) and current Big-M value – if solved successfully move to Step 3.
 - c. Attempt to solve with DICOPT/GAMS and current Big-M value – if solved successfully move to Step 3.
 - d. Relax optimality gap (0.05) and solve with BARON – if solved successfully move to Step 3.
 - e. Relax optimality gap (0.05) and solve with DICOPT/GAMS– if solved successfully move to Step 3.
 - f. Solve with a spatial branch and bound solver – if solved successfully move to Step 3.
 - g. Change Big-M values and return to Step 2a.
3. Solve the non-isocompositional mixing sub-optimisation NLP:
 - a. Use IPOPT (Wachter and Biegler, 2006) with different settings until feasible solution is found.
 - b. If no feasible solution is found, use the MINLP solution with isocompositional mixing to derive unit models.
4. *MassExchanger*: Mass balances from NLP network sub-optimisation are taken to set the boundary values for the detailed packed bed mass exchanger models. Set $fe = 20$. For each exchanger run the following:
 - a. Solve full model exchanger model using fe FEs with IPOPT and CONOPT – if feasible then go to Step 5.
 - b. Solve sequence of simplified NLP models, increasing in complexity, to find feasible initial points to the full model of fe FEs. If full model is solved after initialisations, then go to Step 5.
 - c. Increase fe and go to Step 4a.
5. Derive correction factors based on the differences between the simplified network level solutions and the detailed optimisation models. Check whether current solution of the detailed network is better than current best TAC. If better, store solution as current best. If user selected option to include cuts then derive binary cuts based on current topology and save to MINLP model.
6. Check if the correction factors have changed (within tolerance) between iteration $k-1$ and iteration k . If within tolerance or $k = \text{maximum iterations}$ (set by user), terminate the algorithm and return current best solution (detailed unit models and mass loads). Else, return to step 2 with new correction factors and binary cuts.

This algorithm ensures the code is robust to the many different initial conditions that the algorithm may develop throughout the iterations and removes user involvement. Outputs are generated in the form of a detailed log file with all the iterations as well as various csv files of summarised output. In future versions of the code we hope to include graphical output as well as the incorporation of staged columns, regeneration networks, and HENs.

3. Case studies

To demonstrate the package, we briefly show results from two examples solved using MExNetS. These are available on the Github repository in the examples folder.

3.1 Case study 1

In this case study, H_2S is removed from two gas streams, one (R1) is coke-oven gas, while the other (R2) is tail gas from a Claus unit (Hallale and Fraser, 2000). The problem data is presented in the original paper. To solve this problem in MExNetS, the user needs to decide on a maximum number of iterations, a termination tolerance, a superstructure approach (whether to choose SWS or SBS), and whether to include binary cuts.

Upon running the example with SBS, a tolerance of 0.1 % difference between iterations, and without binary cuts, a solution of \$ 483,691 is found, with termination at iteration 30 in 267 CPUs (including all NLP solves, MINLP solves, and model building/writing and saving) on an Intel Core i7-4700MQ 2.4 GHz PC with 16 GB RAM. The solution is shown in Figure 2a and Table 2. This solution is found in significantly less time than the previous method in Short et al. (2018), and finds a reduction in TAC of 1,582 \$/y. This compares favourably with other best solutions that did not include fixed costs or detailed models as well (\$ 363,691).

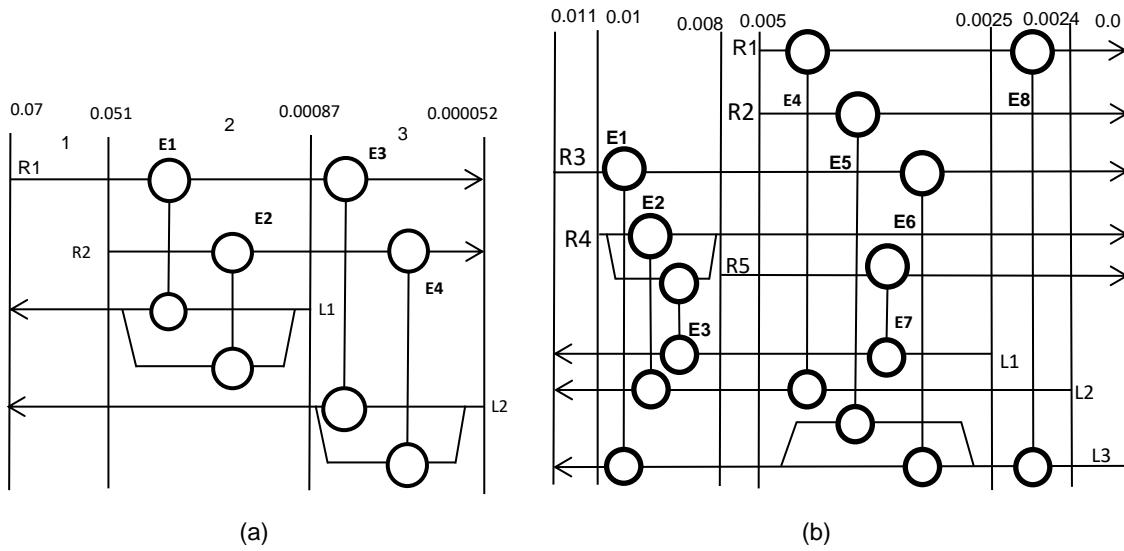


Figure 2: (a) The optimal network for case study 1, (b) Optimal network for case study 2

3.2 Case study 2

This problem, from Short et al. (2018) and adapted from Hallale (1998), involves ammonia removal from five gaseous streams that are predominantly air by two process and one external MSA. The problem data is omitted due to space restrictions. This larger problem takes more time to solve and is more difficult to converge to optimality, and cuts are employed. MExNetS terminates after 42 iterations with an optimal solution found for the detailed MEN found in iteration 26 to be \$ 298,798, 2.8 % lower than the solution found in Short et al. (2018).

Table 2: The optimal packed column designs for the case studies

	Case study 1				Case study 2							
	E1	E2	E3	E4	E1	E2	E3	E4	E5	E6	E7	E8
Height (m)	5.78	1.88	2.06	1.41	4.26	1.84	1.32	0.49	1.19	3.0	2.73	1.96
Diameter (m)	0.72	0.37	0.71	0.32	1.31	0.74	0.59	0.98	0.60	1.39	1.31	0.98
Mass duty (kg/s)	0.06	5×10^{-4}	6×10^{-4}	9×10^{-5}	0.02	5×10^{-3}	3×10^{-3}	10^{-3}	3×10^{-3}	0.01	0.01	0.01
k_y (kg/s/m ³)	0.05	0.03	0.05	0.032	0.05	0.05	0.05	0.05	0.04	0.05	0.05	0.04
Packing size (mm)	36	18	35	16	66	37	30	72	40	70	65	85
Packing cost (\$/m ³)	950	1,519	971	1,610	570	930	1,129	579	947	571	570	696
a_i (m ² /m ³)	129	250	131	285	72.0	126	156	65.5	128	68	72.2	55.8
ΔP (kPa/m)	2.62	5.53	2.69	6.40	1.37	2.56	3.26	1.23	2.61	1.3	1.35	1.03

4. Conclusions

This paper presents an open-source package for optimal simultaneous MENS in Python. To the knowledge of the authors, this is the first open-source non-Pinch Technology-based software package for MENS. The strength of the package lies in its robust initialisation schemes, ease of scripting, integration with detailed models, and ability to overcome local minima and infeasibilities through automatically utilising several optimisation algorithms. MExNetS automatically generates different superstructures based on user choices and can be employed with or without binary cuts to speed solution times. The new package allows for ease of testing and experimentation with the methods employed, as opposed to previous versions of the method that included several manual initialisation and data porting aspects. The new package is shown to provide solutions as good, or better than other methods in several problems, with no user interference. The package is made available on Github for free download and will be extended to include combined heat and mass exchanger networks, regeneration networks, multi-contaminants, and different exchanger types in the future. The authors hope that, through the introduction of such tools and with the transparency of open-source deployment, that the package may stimulate future development in MENS technology.

Acknowledgements

Assoc. Prof. Isafiade would like to thank NRF fund number 87744.

References

- Azeez O.S., Isafiade A.J., Fraser D.M., 2013, Supply-based superstructure synthesis of heat and mass exchange networks, *Computers and Chemical Engineering*, 56, 184–201.
- Chen C., Hung P., 2005, Simultaneous synthesis of mass exchange networks for waste minimization, *Computers and Chemical Engineering*, 29 (7), 1561-1576.
- El-Halwagi M.M., Manousiouthakis V., 1989, Synthesis of mass exchange networks, *AIChE J.*, 35, 1233–1244.
- El-Halwagi, 1997, Pollution Prevention through Process Integration: Systematic Design Tools, Academic Press, San Diego, USA.
- Hallale N., Fraser D.M., 2000, Capital and total cost targets for mass exchange networks, Part 2: Detailed capital cost models, *Computers and Chemical Engineering*, 23, 1681–1699.
- Hart W.E., Laird C.D., Watson J.-P., Woodruff D.L., Hackebeil G.A., Nicholson B.L., Siirola J.D., 2017, Pyomo - Optimization Modeling in Python, Springer International Publishing, Cham, Switzerland.
- Isafiade A.J., Fraser D.M, 2008, Interval based MINLP superstructure synthesis of mass exchange networks, *Chemical Engineering Research and Design*, 86(8), 909–924.
- Isafiade A.J., Short M., 2019, Review of Mass Exchanger Network Synthesis Methodologies, *Chemical Engineering Transactions*, 76, 49-54.
- Isafiade A.J., Short M., 2016, Synthesis of mass exchanger networks for single and multiple periods of operations considering detailed cost functions and column performance, *Proc. S & Env. Prot.*, 103, 391-404.
- Ng D.K.S., Chew I.M.L., Tan R.R., Foo D.C.Y., Ooi M.B.L., El-Halwagi M.M., 2014, RCNet: An optimisation software for the synthesis of resource conservation networks, *Proc. Safety and Env. Prot.*, 92(6), 917-928.
- Oladoso W.A., Wan Alwi S.R., Manan Z.A., 2020, A new algebraic tool for simultaneous targeting and design of a mass exchange network with stream splitting for sustainable environment, *J. Cleaner Prod.*, 249,119361.
- Papalexandri K.P., Pistikopoulos E.N., Floudas C.A., 1994, Mass exchange networks for waste minimization – a simultaneous approach, *Chemical Engineering Research and Design*, 72, 279–294.
- Short M., Isafiade A.J., Biegler L.T., Kravanja Z., 2018, Synthesis of mass exchanger networks in a two-step hybrid approach including detailed unit designs, *Chemical Engineering Science*, 178, 118–135.
- Short M., Isafiade A.J., Fraser D.M., Kravanja Z., 2016, Synthesis of heat exchanger networks using mathematical programming and heuristics in a two-step optimization procedure with detailed exchanger design, *Chemical Engineering Science*, 144, 372-385.
- Sziklai Z., Farkas T., Lelkes Z., Rev E., Fonyo Z., Kravanja Z., 2006, Fairly linear mixed integer nonlinear programming model for the synthesis of mass exchange networks, *Ind. Eng. Chem. Res.*, 45, 236–244.
- Tawarmalani M., Sahinidis N.V., 2005, A polyhedral branch-and-cut approach to global optimization, *Mathematical Programming*, 103(2), 225-249.
- Viswanathan J., Grossmann I.E., 1990, A combined penalty function and outer-approximation method for MINLP optimization, *Computers & Chemical Engineering*, 14(7), 769-782.
- Wächter A., Biegler L., 2006, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.*, 106, 25–57.
- Yee T.F., Grossmann I.E., 1990, Simultaneous optimization models for heat integration-II. Heat exchanger network synthesis, *Computers and Chemical Engineering*, 14(10), 1165–1184.