

Deterministic Global Optimisation can reduce the overall cost of a heat exchanger network up to 23%

How can a Deterministic Global Optimisation (DGO) solver change the way we do optimisation?

Optimisation professionals know that certain formulations are more powerful than others. Also, posing an optimisation problem in a different way can produce much better solutions, because the new model captures more detail or makes fewer assumptions. Unfortunately, usually, the more powerful the formulation, the less solvable the problem.

Non-convex (MI)NLPs can have multiple solutions. So, how can we reliably compare different formulations? How can we know whether a formulation is even feasible to begin with? Current common practice involves highly educated guessing, reformulation, tedious trial-and-error and certain formulations being avoided altogether, because most solvers cannot solve them in reasonable time.

This need not be the case; by empowering users with tools which can **automatically reformulate**, and **prove global optimality** or **global infeasibility** for real-life MINLPs this time-consuming cycle can end. DGO methods allow us to build such tools.

Among optimisation branches, **DGO** is the **only** one able to **guarantee global optimality** (or prove infeasibility) of general, non-convex, MINLP problems, fundamentally changing how we create and solve optimisation problems.

Using a DGO solver is **the only way to prove a problem is infeasible**; an invaluable tool for modelers, informing them with certainty that their model is incorrect, or that what they are trying to do is impossible.

If the problem is feasible, using a DGO solver is the **only way to guarantee global optimality**, i.e., locating the best solution possible. This cannot be guaranteed using conventional optimisation methods with users spending time tweaking their model

and guessing starting points. After significant time investment, they might feel reasonably safe the solution they found is the best they can do. Even then, many powerful formulations remain out of reach. DGO can replace this tiresome cycle of trial-and-error, and second-guessing, with a **single click of a button**.

DGO does not need definition of starting points. **Octeract's DGO** implementation doesn't even require reformulation. The solver automatically derives variable bounds, eliminates singularities, tests different possible reformulations, picks the best one, and starts solving in native parallel mode. Millions of starting points are tested automatically, but not randomly; DGO's branch-and-bound methods guarantee that local solvers will iteratively gravitate towards the global solution. Proving global optimality is powerful but computationally expensive. **Automatic reformulation** and **massively parallel setups** are vital towards solving problems otherwise unsolvable.

In this case study, we delve into a problem that greatly illustrates what benefits can be harvested from using DGO in a heat exchanger network problem.

What are heat exchanger networks and why are they important?

They are networks used in a variety of industries (oil & gas, power production etc) as a means of heat integration. A heat exchanger network comprises of hot streams that need to be cooled down and cold streams that need to be heated up, hot and cold utility streams (i.e. fuel, refrigerant etc) that have an operating cost and devices called heat exchangers to facilitate this process. For a heat exchanger network to be designed, continuous and discrete decisions need to be made. These can have a significant impact on the economics of process design.

The goal of this process is to determine the **optimal** network of heat exchangers to **minimise** the annual operational cost and annualised investment cost. Traditionally, the design of heat exchanger networks requires significant effort, a large variety of computational tools and trial and error to only achieve a design which results into a suboptimal annualised cost.

Octeract proposes the utilization of advanced software tools for the automatic generation of heat exchanger networks that can **reduce the costs by 23%**.

The situation at hand

Let's assume our problem includes one hot stream (H1), two cold streams (C1,C2), one hot and one cold utility (S1, W1, respectively) which may be utilised at a cost.

Properties of streams and utilities

	T_{in} Temperature at input (K)	T_{out} Temperature at output (K)	F_{cp} Total Heat Capacity (kW/K)	h Heat Transfer Coefficient (kW/m ² ·K)	Cost (\$/kW·yr)
H ₁	440	350	22	2.0	-
C ₁	349	430	20	2.0	-
C ₂	320	368	7.5	0.67	-
S ₁	500	500	-	1.0	120
W ₁	300	320	-	1.0	20

The decisions that need to be made for the network design are:

- 1) The amount of hot and cold utilities to be used and the interactions between them and the hot and cold streams
- 2) The number of heat exchangers needed to facilitate these interactions
- 3) The characteristics of these heat exchangers - i.e. heat exchanging area, connection structure (in series or parallel) - which will determine their investment cost

Conventional solution

How are these decisions made today? The answer is, intuitively, and with the use of optimisation techniques that are based on a trial and error approach.

- 1) For the first decision the goal is to **minimise the usage of the costly utilities** by determining the amount of heat that can be transferred from (to) the hot (cold) streams in order to reduce (increase) their temperature while only considering stream-to-stream interactions. If these interactions are not enough to achieve the needed heat exchange, utilities must be used. It should be mentioned that certain stream-stream or stream-utility interactions may be prohibited due to safety specifications or other case specific rationales.

For our example we see that the heat exchange can happen without utilities and so without the cost incurred from their use:

$$\Delta T \cdot H_1 \cdot F_{cp} \cdot H_1 = \Delta T \cdot C_1 \cdot F_{cp} \cdot C_1 + \Delta T \cdot C_2 \cdot F_{cp} \cdot C_2$$

- 2) For the second decision, it makes sense to think that fewer heat exchangers, will result in lower investment cost. As a rule of thumb:

$$\text{Number of exchangers} = \text{Number of streams} + \text{Number of utilities} - 1$$

For our example this is:

$$\text{Number of exchangers} = 3 + 0 - 1 = 2$$

The amount of heat exchangers used will be decided in relation to their size and their configuration (i.e. heat exchange area, in series or in parallel). This can lead to various equivalent designs for the same number of heat exchangers. So then, which combination is the optimal one?

- 3) Once the number of exchangers is determined, the characteristics of each exchanger (i.e. heat exchange area) and the way they are going to be connected need to be decided. The size of the heat exchange area significantly affects the exchanger's cost. For this decision to be made, the formulation of an optimisation superstructure that simultaneously considers the sizing of the equipment and the order of the interactions is designed. The resulting problem is a large scale non-convex, nonlinear programming

problem (NLP) that attempts to **minimise** investment cost based on the overall heat exchange area.

For our example, the annualised cost has been calculated using the following **nonlinear** equation:

$$\text{Single Heat Exchanger Cost} = 6600 + 670 \cdot (\text{Area})^{0.83}$$

Having solved these three subproblems we end up with the following decisions:

Sequential Solution Table

	1st decision:		2nd decision:	3rd Decision:	Minimise Cost
Solution Type	Number of utilities	Type of utility	Number of heat exchangers	Heat exchanger area (m ²)	Annualised investment cost (\$)
Sequential	0	-	2	HE ₁ : 425.99	121 500
				HE ₂ : 14.96	

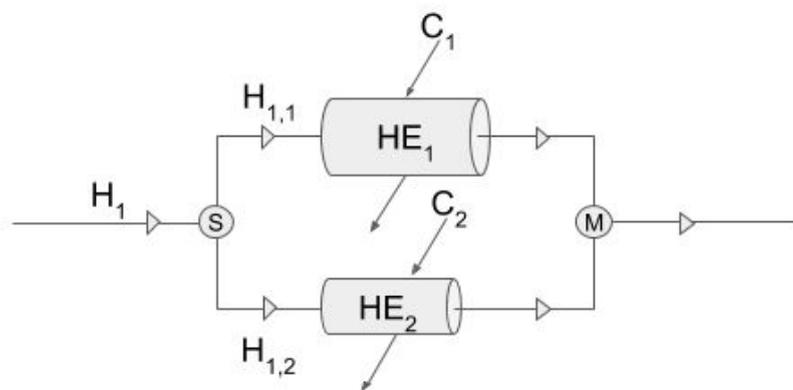


Figure 1: Diagram of heat exchanger network proposed by the sequential solution

The above decisions seem logical enough. So what is the issue here?

- a) Even if it is assumed that the optimal result in each subproblem can be determined, the overall solution may still be suboptimal because the overall network design problem has been decomposed into a series of subproblems.
- b) Solutions to the problem described here that seem intuitively correct, may be suboptimal when optimisation techniques are not applied.

What does **Octeract** suggest?

The **simultaneous optimisation** of the heat exchanger network. All possible and thermodynamically meaningful combinations of heat exchangers are considered at once. The necessary heat exchange area is considered and the investment cost associated with those decisions is annualised and added to the annual operating cost of any utilities necessary to fulfil the heat exchange. The resulting formulation can be a large scale, structurally non-convex, Mixed Integer Nonlinear Programming which until today **posed a significant challenge**.

Solving this problem leads to the decisions outlined below:

Simultaneous Solution Table

	1st decision:		2nd decision:	3rd Decision:	Minimise Cost
Solution Type	Number of utilities	Type of utility	Number of heat exchangers	Heat exchanger area (m2)	Annualised investment cost (\$)
Simultaneous	2	Hot (S_1)	4	HE ₁ : 130.01	
				HE ₂ : 13.77	
		Cold (W_1)		HE ₃ : 4.83	
				HE ₄ : 2.82	93 739

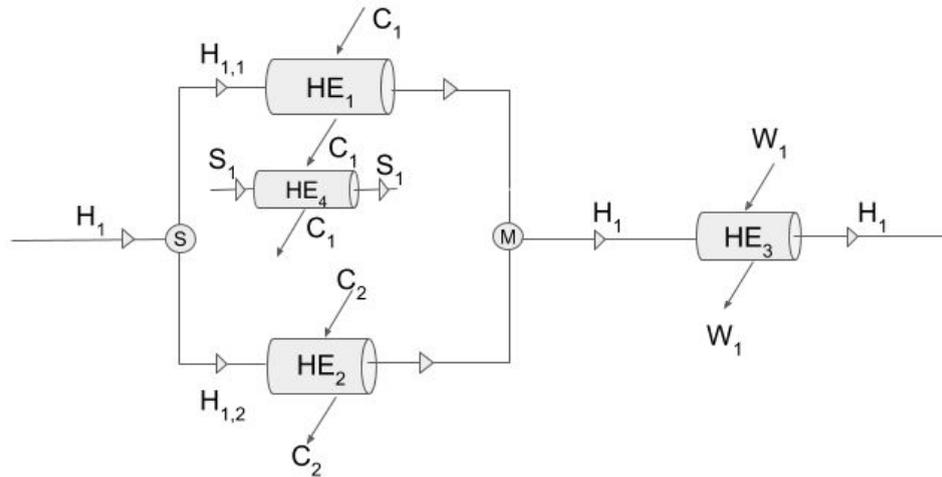


Figure 2: Diagram of heat exchanger network proposed by Oceract's Frontier solver

How is it possible to use utilities and more exchangers and get a lower cost?

The significantly smaller exchange area in the heat exchanger between H_1 & C_1 causes (i) a smaller temperature drop for H_1 and (ii) a much smaller heat exchange cost. The tradeoff is that a much larger heat exchanger is needed for a small increment in the temperature drop. In fact, the required heat exchanger in the sequential approach overcompensates for utility usage and the investment into two small heat exchangers. As a result, in the simultaneous approach, the residual heat from using smaller heat exchangers among the streams can be handled by small utility amounts in small heat exchangers whose overall cost is significantly smaller than the investment in a very large equipment.

What did DGO allow us to do differently?

We **proved** that the simultaneous formulation is **23% better** than the sequential in solving a heat exchanger network problem. Both problems were solved to global optimality in one click when with conventional technology, getting these solutions can be time consuming, challenging, and often impossible. Rigorously comparing these non-convex formulations is possible because of the use of DGO.