THE EFFICIENT MODELLING OF STEAM UTILITY SYSTEMS

Jonathan Currie[1], David I. Wilson[2]

[1]Electrical & Electronic Engineering
AUT University
Auckland, New Zealand
jocurrie@aut.ac.nz

[2]Industrial Information & Control Centre
AUT University
Auckland, New Zealand
diwilson@aut.ac.nz

ABSTRACT

Industrial utility systems, traditionally the "poor forgotten cousin" of the processing industry, are now receiving the attention they deserve. While the thermodynamics and process models of utility systems are well established, the optimization of utility systems typically involves solving large-scale optimization problems including integer constraints, and hence are less commonly tackled. The integer formulation arises naturally in circumstances for example where process flows drop to zero, or processing equipment is brought into, and out of service. Both these cases are far more common when modelling utility systems as opposed to the now routine modelling of the parent chemical processes. This paper investigates various alternative modelling and subsequent optimisation strategies for a large industrial steam utility system. The results show that by applying suitable models, good modelling choices and appropriate software, one can readily solve large problems to obtain significant economic savings in solution times that are even amenable for design and/or real time optimization.

INTRODUCTION

Infrastructure to produce heat and power for the local consumption within chemical complexes is referred to as a utility system. The most common, and arguably the most efficient, are systems that use steam as the working fluid. Given the scale of these systems, the potential for either saving energy consumption, or better utilisation of what is produced is considerable. Small changes in operating point, the swapping into, and out of service of major equipment all can have a large, and at times non-intuitive economic effect (Bruno, Fernandez et al. 1998; Eastwood and Bealing 2004; Aguilar, Perry et al. 2007). Once the utility system grows beyond a modest size it becomes practically impossible for a human operator to establish the global optimum with using a carefully crafted, mathematically based search technique, such as in (Rodríguez-Toral, Morton et al. 2001; Strouvalis, Heckl et al. 2002; Lee and Grossmann 2003).

This paper is concerned with the task of first modelling steam utility systems, and then the subsequent optimisation of these large, multi-faceted, interacting systems. We show that the introduction of discrete operations, something quite common in utility systems, the problem becomes non-convex and is plagued by the possibility of local minima (Lee and Grossmann 2003). However good modelling techniques coupled with a suitable environment calling on large-scale optimisation algorithms can result in considerable savings that otherwise might elude the system operator. In some cases using global optimisation strategies, we can be sure of the optimum. What is interesting is that unlike the general nonlinear, mixed-integer optimisation problem, we show that one can pose and solve for the optimal point in utility systems in a reasonable time using commonly available code for cases that are of industrial significance.

UTILITY SYSTEM CASE STUDY

This paper considers the steam utility system shown in Figure 1. Although hypothetical, this example is representative of industrial systems at a reasonable level of complexity.

The utility system consists of three steam headers at 40, 11 and 4 bar, supplied by three boilers; a small (20 tonne/hr), medium (40 tonne/hr) and a large (60 tonne/hr) boiler. Two backpressure turbines supply shaft work to the plant. Both turbines contain parallel electric motors for redundancy. The shaft work requirements are for 600 and 200kW for the HP (high pressure) and MP (medium pressure) turbines, respectively. In addition, two 2000kW turbogenerators are installed which can generate electricity for the site. Two fixed steam users representing process heat demands and desuperheaters between headers are used to meet the steam demand of the lower headers. Finally a condensate collection system and flash to recover useful energy from the steam users completes the system loop.

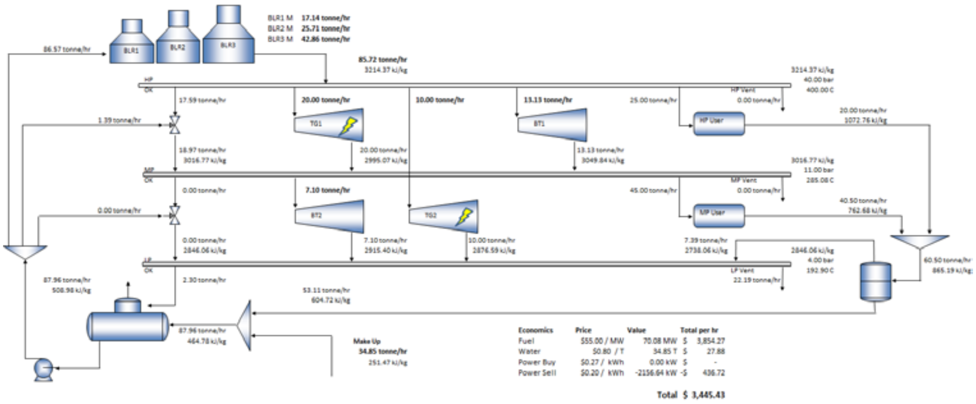

Figure 1: Base case utility system using the JSteam Excel Add-In.

This steady-state model was built using the JSteam Excel add-in (Currie 2012), an Excel add-in that our research team has developed. It is based on our industrial experience in utility system modelling and simulation, and has been validated by international petrochemical utility systems (Currie, Wilson et al. 2011).

The package contains high speed rigorous thermodynamics based on the International Association for the Properties of Water and Steam tables, IAPWS (2007) and a suite of utility system unit operations, allowing detailed utility systems to be quickly modelled and Operational Expenses (OPEX) estimated. The software is available as a free demo, from our website, www.inverseproblem.co.nz and includes all functionality used within this paper.

Base Case

The base case operating scenario illustrated in Figure 1 shows the current operating point of the utility system, together with a breakdown of the hourly cost of operation.

In this nominal operating scenario, the system is meeting the heat requirements of the process, as well as providing shaft work via both backpressure turbines. The operator has also opted to generate 2150kW of electricity via the steam turbo generators, which can be sold back to the grid to reduce OPEX. As all three boilers are already running, the steam demand is split between them.

A characteristic of utility systems is that small changes in the operating point can result in significant OPEX savings. Using the JSteam modelling environment, the user can examine potential savings by manually modifying the operating conditions of plant equipment, perhaps using an enlightened trial and error approach to search for economic savings. However for a real-time approach, or for larger interconnected systems which is the focus of this research, an automated strategy using mathematical optimization techniques is the only practical method. This paper describes various modelling and optimization strategies in the following sections~~detail~~.

MODELLING AND OPTIMIZATION

Our chosen modelling and optimization environment for this research is MATLAB. This is based on the rapid prototyping nature of the language and our development of the large-scale optimization toolbox, OPTI (Currie and Wilson 2011). OPTI collects together a suite of open source linear, nonlinear and discrete optimizers within MATLAB, all with a common calling interface which enables one to evaluate a range of models, structures and optimizers for this research. The functionality included in the JSteam Excel add-in has also been ported to MATLAB for this research.

Utility System Models

The simulation model uses unit operation models we developed while consulting utility system modelling, as well as previously published literature models. Simple unit operations such as desuperheaters, pumps and flash vessels are described using standard mass and energy balances, together with knowledge of the operation of the equipment. However the boiler, turbogenerator and backpressure turbine models require regressions based on published data in order to estimate the performance given a range of equipment sizes and part load operation.

As shown in early work by (Peterson and Mann 1985) who surveyed a range of turbines over varying input conditions, the maximum achievable efficiency (at full load) is a nonlinear function of the equipment size. This relationship was used in (Varbanov, Doyle et al. 2004; Aguilar, Perry et al. 2007) to build both nonlinear and linear models of backpressure and condensing turbines. We have derived the turbine models in our modelling framework from this work; however we have modified them to better approximate the turbine behaviour as it approaches a zero flow (off) condition in order to assist the optimizer.

The boiler model is based on work by (Shang and Kokossis 2000) and the revised model in (Aguilar, Perry et al. 2007), and again modified to better approximate zero flow conditions. Together with the modified turbine model, our models allow for varied equipment sizes, part load performance, and expected operation near zero flow conditions. It should be noted that the development and validation of the utility models is not the focus of this paper, (the interested reader can review the above mentioned papers for further details in this aspect), but rather the optimisation step after a reasonable model has been developed.

Optimizing the Simulation Model

An obvious but naïve approach is to simply wrap the existing simulation model inside a general constrained nonlinear optimizer. This approach requires the least amount of problem formulation and one would expect it to be the most accurate approach, given the high fidelity of the simulation model.

In the current implementation, the utility model is built in MATLAB using a sequential-modular approach where the information flow is in one direction through the model.

Objective Function

The cost function supplied to the optimizer considers three components of the utility system; the fired duty of the boilers, electricity usage / generation, and demineralised water use. The resulting expression is detailed in equation (1):

$$j = \alpha \sum_{n=1}^{3} \mathrm{Blr}Q_n + \beta \left(\sum_{n=1}^{2} \mathrm{TG}Q_n - \sum_{n=1}^{2} (1 - b_n)\mathrm{BPT}\,Q_n \right) + \gamma \mathrm{WaterM} \qquad (1)$$

where α is the cost of boiler fuel gas per MW, β is the cost of buying or selling electricity (dependent on the site electricity balance) per kWh, and γ is the cost of demineralised water per tonne. The turbine terms are split via machine type; turbo generators are priced directly by the shaft work power they produce, while back pressure turbines are priced whether they are on ($b_n = 1$) or off, inferring they require electricity from the site supply. Note that for this work we have assumed all boilers run on the same natural gas supply, therefore the fired duty can be directly related to fuel gas consumption via the Higher Heating Value (HHV) of the fuel, and therefore priced via α.

The two binary terms, b_1 and b_2, will be fixed to 1 (i.e. equipment is switched on) for the first part of this paper, to allow investigation of continuous optimization of these systems. In the second part, these binary variables will be free decision variables.

Approximating Gradients

As with all typical utility and process flow models there are several recycle loops present in the system (Currie, Wilson et al. 2011). These occur due to condensate collection, desuperheater cooling water, as well as the overall system loop, all of which ensure a system energy balance.

The issue with a recycle is that it cannot be solved directly using a sequential modular approach (or a non-sequential modular approach such as used by the commercial packages HYSYS or VMGSim). The system must be iterated using a multi-variable root solver to converge and this results in two nested optimization loops, the inner as the root solver (typically posed as an optimization problem itself), and the outer as the nonlinear optimizer used to minimise the operating cost.

While it is not uncommon to have nested optimization loops (for instance most optimizers solve at least one internal sub-optimization problem as part of each iteration), it is unlikely the optimizer converges quickly, if at all, when the outer loop relies on a gradient (first and for some solvers, second derivatives) obtained via finite-differences of the inner loop. This is due to the decreased accuracy of the derivatives based on non-deterministic properties of the inner loop, which the optimizer uses to establish convergence.

Analytical gradients are not available when using a modular simulator model (especially with a thermodynamic engine), so one must approximate them using a finite-difference approach, or use a gradient free optimizer. For this example we are using the gradient based nonlinear optimizer IPOPT (Wachter and Biegler 2006), which means that derivatives will be approximated. The inner multi-variable root solver is NL2SOL, a nonlinear-least-squares solver adapted for nonlinear equation solving.

Decision Variables

An advantage of utilizing the simulation model is the reduction of decision variables as typically only a few mass flows and states are required for the objective calculation. For this model seven variables have been selected. These include two variables for the mass flows through the turbo generators, three variables for the steam production fractions (of the total steam demand) of each of the boilers and two binary variables representing the state of each backpressure turbine (on / off). As mentioned for the initial case study, the binary variables in this instance have been fixed to 1.

Nonlinear Constraints

In order to ensure a system mass balance and keep mass flows within machine limits, six nonlinear inequality constraints are used. Three constraints constrain the steam production through the boilers, two constrain the power production by the turbo generators, and one constrains the HP header vent to be greater or equal to zero. This constraint is required to ensure HP steam is provided via the boilers and not via a negative mass flow on the vent. As we have not bounded any mass flows to be greater or equal to zero (to reduce decision variables), this constraint ensures a sensible system mass balance.

In order to implement the above constraints, the simulator model needs to be solved again for each constraint evaluation. This infers that the entire model must be converged again (using the multi-variable root solver) for every objective, gradient, constraint and

Jacobian evaluation. The result is a computationally expensive and inefficient optimization run.

Optimized Simulator Model

Using the base case model in Figure 1 described in MATLAB and optimized using IPOPT, the hourly cost reduces 15.13% from \$3444/hr to \$2923/hr, taking 35 seconds to converge on a 64bit Intel i7-920. The resulting operating point is shown in Figure 2.

The optimizer has made three major changes to the base case operating point. It has reduced the steam mass flow through turbo generator 2 to zero, reducing steam vented via the low pressure (LP) header, as well as increasing steam flow via turbo generator 1 to maximum. It has also reallocated steam production between the boilers to maximize firing efficiency.

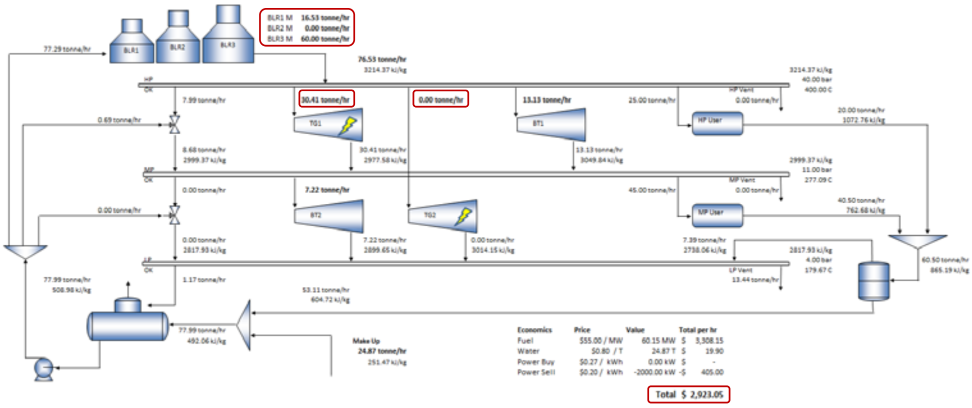

Figure 2: Optimized operating point using the simulator model.

Figure 3 shows the simulation model (Sim) has improved the bottom line OPEX by 15%. However we can further improve on those savings which will be detailed in the succeeding sections.

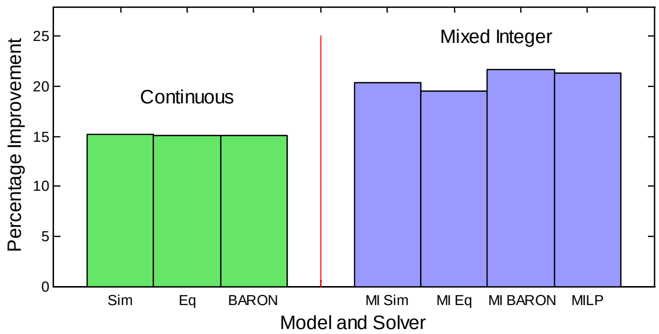

Figure 3: Percentage improvement for the strategies discussed in this paper.

Optimizing Using a Special-Purpose Model

An alternative to optimizing the generic simulator model is to build a new model designed specifically for optimization. This involves opening up the black box unit operations and writing the entire system in terms of a mass and energy balance. For utility systems this is quite simple given the single component system (few thermodynamic calculations required) and simple models with few internal equations.

It is important to consider at this point that if one undertakes the decision to build an optimizer model, you now have *two independent* models for the utility system. This requires disciplined management in order to maintain both models based on the current physical system and operating conditions. As part of the JSteam software framework we are working on functionality which will automatically convert the JSteam simulator model into an optimization suitable model.

System Mass Balance

The system mass balance can be written entirely in terms of linear equality constraints. However this requires that every stream (represented by an arrow in Figure 1) contains a decision variable representing the mass flow of that stream. For the case study model this requires 44 decision variables.

The model mass balance is described by 29 linear equality constraints, and by using OPTI, these will automatically be used to generate the constraint Jacobian (eliminating the need for a finite difference approximation).

System Energy Balance

While the mass balance is linear, the system energy balance is at best bilinear, and at worst nonlinear. General equipment (such as desuperheaters, mixers and headers) can be described using bilinear equality constraints, noting that bilinear terms exist due to the multiplication of mass flow variables and enthalpy variables.

In order to more accurately model the backpressure turbines and turbo generators, nonlinear efficiency curves are implemented. These infer general nonlinear constraints, however, as discussed, these are implemented so that the transition between zero flow and operational does not result in unrealistic values.

The model energy balance is described by 10 bilinear and 6 nonlinear equality constraints. Ten extra decision variables are implemented to represent the enthalpy of streams around the two lower headers. This allows the header enthalpies to vary depending on equipment output enthalpies to more accurately represent the system.

Two further nonlinear inequality constraints are placed on the turbo generator output powers to ensure machine limits are met.

Objective Function

The objective function implements explicit boiler and turbo generator models which calculate the fuel duty and generated power respectively. An electricity balance ensures power is either imported or exported as required, as well as being bought and sold at respective prices. The result is a separable nonlinear expression which adds to the total hourly cost of the system.

As with the simulator model, two binary variables representing the state of the backpressure turbines are included, but are still fixed to 1 with this scenario.

Optimized Equation Based Model

The final equation-based model contains 56 decision variables and 159 constraints. The constraints are made up of 112 bounds (2 for every decision variable), 29 linear equality, 16 nonlinear equality and 2 nonlinear inequality constraints. Given that the intention of this paper is to compare optimisation strategies, we have decided to report the metrics pertaining to the size of the optimisation problem (the number of variables and constraints etc), which at a first approximation, give an indication to the complexity of the problem. Without actually laboriously listing the myriad of equations, this is probably the next best option.

Given the large increase (8x) in both decision variables and constraints it could be easy to assume that the solution time for this optimization problem will be much slower. However given that IPOPT is a large-scale Nonlinear Program (NLP) solver (and this problem is nowhere near large-scale), it performs well, solving the same optimum of $2923/hr in just 0.34 seconds. This is a 100x speed improvement for a problem which is 8x bigger, and based on the comments by our industrial partners, of extreme practical significance. The operating point is the same as described in Figure 2 and detailed in Figure 3.

Implications of Integer Constraints

Both models have so far included binary variables representing the state of the backpressure turbines, namely on or off. When the turbine is on, steam is used to drive the process shaft work demand (taken to be a fixed duty), while when off, an electric motor provides the shaft work. A common, but deceptively complex optimization problem is to find which is the more economical; generating steam to drive the turbine, or paying for electricity.

In both scenarios described so far we have artificially fixed these turbines on. This enables us to solve these problems as continuous nonlinear programs, which are much easier to solve than Mixed Integer NLPs (MINLPs). If we wish to be able to manipulate the turbine state and remain smooth (i.e. without using rounding functions) the problem must be cast as a MINLP.

To solve the resulting MINLPs we use the OPTI included solver BONMIN (Bonami, Biegler et al. 2005; Bonami, Biegler et al. 2008), which implements a branch and bound framework utilizing IPOPT as the relaxed solver.

MINLP Simulator Model

The simulator model is resolved using BONMIN and a new optimum of $2741/hr is obtained in 203 seconds. The new operating point is shown below:

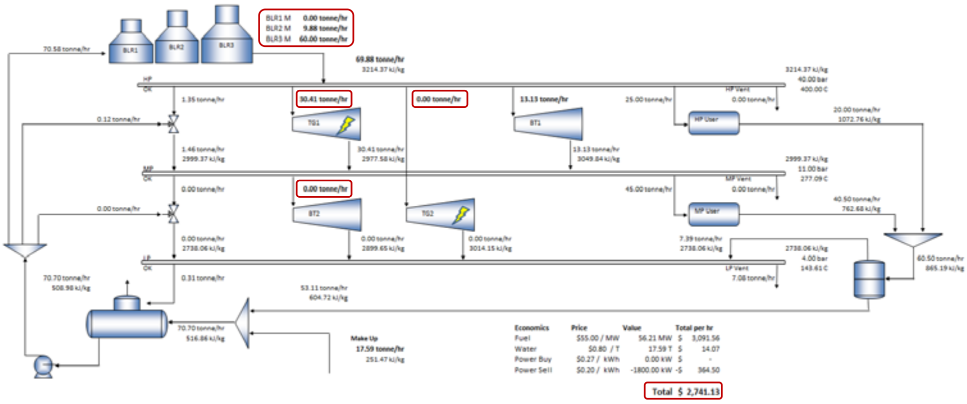

Figure 4: Optimized operating point using a MINLP optimizer and the simulator model

The mixed-integer optimizer has made two changes from the original optimized point resulting in a further $180/hr saving. As shown in the figure above, it has switched backpressure turbine 2 off as well as reallocating the steam production between the boilers. It is worth noting at this point that it has not optimally redistributed steam production (Boiler 1 would be more efficient at that demand instead of Boiler 2), thus it appears to have found a local minima.

MINLP Equation Model

The equation based model is also resolved using BONMIN and it returns an optimum of $2770/hr in 3.3 seconds. This time the optimizer has optimally reallocated the steam production between the boilers, but has turned off both back pressure turbines. While this reduces overall steam demand (and thus fuel gas cost), it does so at a cost of exporting less electricity to the grid. This combination actually results in a higher hourly cost. It appears that this result is also a local minima.

Global Optimality using White Box Models

So far we have demonstrated solving a mixed integer nonlinear utility system using two modelling approaches: sequential modular (the simulator model) and equation based modular (the optimizer model). There is an important similarity (and disadvantage) between both approaches presented so far: they both rely on black box models. This means the optimizer cannot exploit problem structure when solving the problem, and instead views the model as an unknown black box with unknown problem properties.

Exploiting Problem Structure

An example of where problem structure can be useful, and is not currently exploited, is the way IPOPT handles linear and nonlinear constraints. Using the current IPOPT interface, it will only accept *all* linear inequalities or equalities. If one constraint is nonlinear, then the entire constraint Jacobian will be evaluated for each call, even though all linear constraint elements will be static.

A further benefit of knowing the problem structure is the ability to apply automatic differentiation to calculate exact first and second derivatives. This is well known to provide large performance and convergence benefits for all gradient based optimizers.

Searching for Global Optima

Shown in the results presented thus far are two different optima returned by each model. In both cases the optimizer determined the value returned to be optimal. This is due to both IPOPT and BONMIN being local (convex) optimizers only. Both solvers will attempt to solve non-convex problems, but will only return the first local solution.

Using process knowledge when examining the base case, it is almost trivial to pick the global optimum, for which neither model has solved. There are further savings we can make while still substantially decreasing OPEX. In order to reliably find the global minimum using an optimizer, we cannot use IPOPT or BONMIN in this operating scenario.

Global Optimization using BARON

A recent development by our research team is a MATLAB interface to the white box global solver BARON (Tawarmalani and Sahinidis 2005). Our research team worked together with the BARON developer to develop the interface and it has been applied to our utility system optimization research.

To use BARON instead of BONMIN, no model changes are required. The objective and constraints are passed without change to BARON, and the interface automatically generates and solves a white box implementation of the model, to proven global optimality. This problem is especially suited to BARON because it recognises and exploits both bilinear and convex terms automatically, and generates problem tree cuts to remove sub-problems which don’t need to be solved. This means it is a very efficient solver for problems with bilinear and quadratic constraints, such as the energy balance in this utility model.

Solving with BARON returns an optimum of $2696/hr in 2.4 seconds. This matches the solution by inspection, which enables backpressure turbine 1, and optimally allocates steam production between boilers 1 and 3.

Reformulation as a Mixed Integer Linear Model

A common thrust in this research area is the formulation of the utility model as a Mixed Integer Linear Program (MILP). This has the advantage of solving much larger problems, including those with multi-period based objectives. Being linear, the optimizer can also exploit problem structure, which is typically done via pre-solving and cut generation. The downside to a linear model is the approximations of system characteristics needed in order to keep the system linear.

Linearizing the Objective

In previous MILP utility system optimization, such as (Aguilar, Perry et al. 2007) the boiler is viewed as a linear relationship between fuel consumption and steam production. While this is approximately true, as viewed in the left diagram of Figure 5, there is still sufficient error to warrant a piecewise approximation. This is based on the fuel cost being a major component of the system cost, and thus should be modelled more rigorously.

Within the JSteam framework the cost of running boilers and turbines can be automatically approximated using an optimal piecewise linear approximation. This is derived by posing the problem as a nonlinear least squares problem and solving for the intercept points. The resulting intercept points are then converted into a Special Ordered Set (SOS) of type 2 and augmented to the problem to be used with the MILP solver.

Using JSteam each of the 3 boilers and 2 turbo generators are therefore characterised in terms of the steam mass flow versus operating cost / electricity price and piecewise approximations generated.

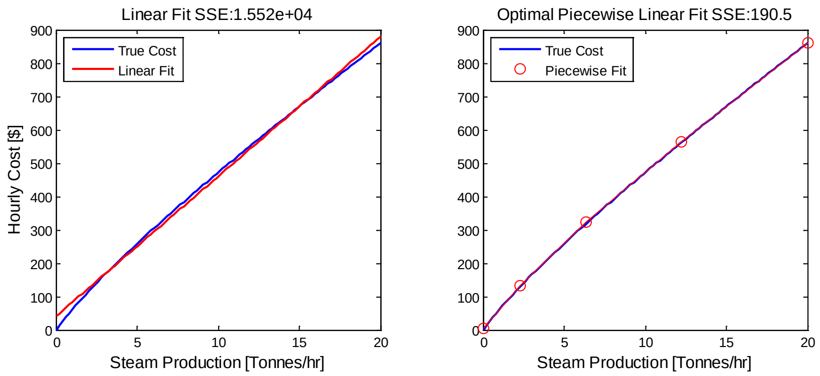

Figure 5: Comparison of linear fit versus piecewise approximation to boiler 1 running cost.

The two remaining cost elements are the demineralised water cost, and to complete the electricity balance. The demineralised water cost is a linear relationship to mass flow, while the electricity balance requires an extra four binary variables to be introduced. These represent either-or conditions for each of the backpressure turbines, allowing the linear objective to calculate the sum of imported and generated electricity at the correct price.

Linearizing the Constraints

Given that mass balance equations are linear it is only the energy balance equations which must be linearized. In order for all constraints to remain linear, the nonlinear terms must be either approximated using linear regressions or piecewise linear fits, or constants need to be introduced.

The most common method in literature is to fix the header enthalpies prior to optimization. This replaces most of the energy balance bilinear terms with linear equality constraints. It is also possible to pre-calculate many enthalpy terms of the system which do not vary with mass flow, such as around the condensate collection, deaerator and pump. The remaining energy balance constraints can then be written as linear equations of these constants.

Equipment limits, such as those required by turbo generators are automatically constrained via the use of SOS.

Solving the Resulting Linear Model

The resulting mixed integer linear model contains 73 decision variables and 202 constraints. The constraints are made up of 146 bounds, 47 linear equality constraints, 5 special ordered sets of type 2, and 4 explicit binary variables. The problem can be solved using CBC, an open source MILP solver with SOS support and supplied with OPTI. Solving with CBC returns an optimum of $2711 in 0.04 seconds.

The operating point solved is effectively the same as that obtained by BARON, however the slightly increased hourly cost is due to the MP header enthalpy not being calculated, which requires a slight increase in steam production.

An interesting result was that if the boiler running cost curves were approximated by only a linear regression, a sub-optimal operating point was returned by CBC with a different allocation of steam production between the boilers.

SUMMARY OF RESULTS

Table 1 summarises the optimization results obtained in this paper and compares key system variables. From an asset owner/operator's perspective, the cost row in Table 1 is the most important result; this shows the potential savings. From a modeller's perspective, the solution time is also of high interest. Indeed for systems know to be complex, or if one is using a global optimisation strategy such as BARON, then the solution time is the key result.

Table 1: Comparison of results versus modelling and optimization strategy.

Property /Variable	**Base Case**	**NLP Sim Model**	**NLP Eq Model**	**NLP BARON**	**MINLP Sim Model**	**MINLP Eq Model**	**MINLP BARON**	**MILP**
Cost [$/h]	$3444	$2923	$2923	$2923	$2741	$2770	$2696	$2711
Solve Time [s]	0.04	35	0.34	0.5	203	3.3	2.4	0.04
Iterations	0	56	25	24	0	0	11	4
TG1 Q [kW]	1218	2000	2000	2000	2000	2000	2000	2000
TG2 Q [kW]	938	0	0	0	0	0	0	0
BPT1 M [T/h]	13.13	13.13	13.13	13.13	13.13	0	13.13	13.13
BPT2 M [T/h]	7.1	7.22	7.23	7.23	0	0	0	0
BLR1 M [T/h]	17.14	16.53	16.54	16.54	0	8.72	9.88	9.89
BLR2 M [T/h]	25.72	0	0	0	9.88	0	0	0
BLR3 M [T/h]	42.86	60	60	60	60	60	60	60

CONCLUSIONS

This paper has compared four different modelling and optimization strategies with respect to a simple utility system model. Using MATLAB with JSteam and OPTI we have demonstrated how wrapping the existing simulator in an optimizer may return the correct minimum, but at a significant performance time penalty. Moreover, when introducing binary and integer constraints into the model, the proposed utility model exhibits multiple local minima. This is attributed to the non-convexity of the nonlinear equality constraints which are known to require a global solver. By utilizing the white-box global optimizer, BARON, we were able to use the existing models and solve them to global optimality with minimal optimization time, and minimal model building time. It was also shown that a linearized model with piecewise linear approximations to nonlinear characteristics obtained the global minimum in the least amount of time; however it also required the most amount of time to develop.

ACKNOWLEDGMENTS

Financial support to this project from the Industrial Information and Control Centre, School of Engineering, AUT University and The University of Auckland, New Zealand, is gratefully acknowledged.

REFERENCES

Aguilar, O., S. J. Perry, et al. (2007). "Design and Optimization of Flexible Utility Systems Subject to Variable Conditions: Part 1: Modelling Framework." Chemical Engineering Research and Design **85**(8): 1136-1148.

Aguilar, O., S. J. Perry, et al. (2007). "Design and Optimization of Flexible Utility Systems Subject to Variable Conditions: Part 2: Methodology and Applications." Chemical Engineering Research and Design **85**(8): 1149-1168.

Bonami, P., L. T. Biegler, et al. (2005). An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs, IBM Research Report.

Bonami, P., L. T. Biegler, et al. (2008). "An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs." Discrete Optimization **5**(2): 186-204.

Bruno, J. C., F. Fernandez, et al. (1998). "A Rigorous MINLP Model for the Optimal Synthesis and Operation of Utility Plants." Chemical Engineering Research and Design **76**(3): 246-258.

Currie, J. (2012). JSteam Excel Add In, Industrial Information & Control Centre**:** An Excel Add In for Utility System Modelling.

Currie, J. and D. I. Wilson (2011). OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. Foundations of Computer-Aided Process Operations (FOCAPO) 2012, Savannah Georgia.

Currie, J., D. I. Wilson, *et al*. (2011). Steam Utility Systems are not 'Business as Usual' for Chemical Process Simulators. American Institute of Chemical Engineers (AIChE) Spring Meeting, Chicago, USA.

Eastwood, A. and C. Bealing (2004). "Optimizing the Day-to-Day Operation of Utility Systems." Energy Engineering **101**(1): 7-25.

IAPWS (2007). Revised Release on the IAWPS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, International Association for the Properties of Water and Steam**:** 49.

Lee, S. and I. E. Grossmann (2003). "Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks." Computers & Chemical Engineering **27**(11): 1557-1575.

Peterson, J. F. and W. L. Mann (1985). "Steam system design: how it evolves." Chemical Engineering **92**(21): 62-74.

Rodríguez-Toral, M. A., W. Morton, et al. (2001). "The use of new SQP methods for the optimization of utility systems." Computers & Chemical Engineering **25**(2-3): 287-300.

Shang, Z. and A. C. Kokossis (2000). Analysis and optimization of total site utility systems. UMIST, University of Manchester. **PhD**.

Strouvalis, A. M., I. Heckl, et al. (2002). "An accelerated Branch-and-Bound algorithm for assignment problems of utility systems." Computers & Chemical Engineering **26**(4-5): 617-630.

Tawarmalani, M. and N. Sahinidis (2005). "A polyhedral branch-and-cut approach to global optimization." Mathematical Programming **103**(2): 225-249.

Varbanov, P. S., S. Doyle, et al. (2004). "Modelling and Optimization of Utility Systems." Chemical Engineering Research and Design **82**(5): 561-578.

Wachter, A. and L. T. Biegler (2006). "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming." Mathematical Programming **106**(1): 25-57.

Formatted: Font: Italic