17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

Neural and genetic based techniques for solving the MSF model as opposed to conventional numerical methods

Farouq Mjalli[a], Nabil Abdel-Jabbar[b], Hazim Qiblawey[c] , Hisham. Ettouney[d]

a Dept. of Chem. Eng., University of Malaya, Malaysia, E-mail: farouqsabri@yahoo.com
b Dept. of Chem. Eng., American University at Sharjah, UAE, E-mail: nabdeljabbar@aus.edu
c Dept. of Chem. Eng., Jordan University of Science and Technology, Jordan hazim@just.edu.jo
d Dept. of Chem. Eng., Kuwait University, Kuwait, E-mal: ettouney@hotmail.com

Abstract

Simulation of multi-stage flash (MSF) desalination processes that have production capacity range between 50,000 to 75,000 m^3/d is an intensive computational problem that requires high computer processing speed despite the availability of advanced processing computer power in hand nowadays. In this work, a comparative study is conducted to explore the performance of different numerical techniques to solve large set of nonlinear equations generated by large scale MSF models. These algorithms can be categorized into three groups namely; conventional numerical approximation methods, multi-objective optimization based methods, and the last group comprises artificial neural networks (ANN) based models and genetic algorithms (GA) based methods. The problem of solving large sets of nonlinear equations with upper and lower constraints is accomplished successfully using all algorithms with different prediction efficiency and speed. The idea of using GA and ANN based algorithms in simulating the MSF model is basically used to generate feasible initial solution estimates that were used as starting guesses for other numerical methods in the former case and to eliminate the step of providing these initial guesses in the later case. Significant reduction of computation effort was

attained using ANN-based techniques. The outcome of this work can be utilized to develop new generation of process simulators that are based on well trained ANNs in order to achieve speedup of computations and to generate more reliable predictions without detracting from accuracy.

Keywords: MSF, Genetic Algorithms, ANN, optimization, simulation.

1. Introduction

The multistage flash desalination (MSF) accounts for 50% of the entire desalination industry. Since the late 1950's the process capacity increased from less than 500 m^3/d to 75,000 m^3/d. The problem of solving the MSF model was considered in many previous studies using different mathematical approaches. Glueck et al. [1] and Hayakawa et al. [2] applied the Newton-Raphson procedure to solve the model. Helal et al. [3] linearized the system of nonlinear equations and formulated a tridiagonal matrix that was solved using Thomas algorithm. El-Dessouky H., Bingulac [4] used a fixed point iterative algorithm to solve the same problem. Others approached the MSF model problem solution using optimization techniques [5-7]. Despite the efficiency of these methods, they remain specific to the system of equations considered and cannot be generalized for all process models. Evolutionary based optimization techniques such as the Genetic Algorithms (GA) have been used extensively over the past few years. Genetic Algorithms has the advantage of solving optimization problems without the need to compute function derivatives. Another advantage is that GA algorithms do not require an initial condition to converge to an optimal solution. Artificial Neural Networks (ANN) are well known for their excellent property of approximating the behavior of processes. They have been the focus of many system identification studies in the past two decades [8-11].
From the previously mentioned work in the field of MSF model simulation, a great need stems for finding a more reliable solution method that exhibits properties like fast convergence, assured system stability, high modeling accuracy and reduced computational overhead. In this work, a comparative study is conducted to explore the merits and shortcomings of conventional solvers as compared to more advanced techniques in solving the MSF model.

2. MSF Process and Model

The MSF desalination process consists of (n) stages, the brine heater, the vacuum ejector, the condenser, chemical addition pumps, and the freed screens. Flashing takes place from the brine flowing across the stages. The flashed off vapor condenses on the condenser tubes, where the feed seawater recovers the latent heat of condensation before entering the brine heater [12]. The MSF model contains balance equations for each flashing stage as well as correlations for physical properties and heat transfer coefficients. Assumptions used to

develop the model include the following: Steady state operation; Heat losses to the surroundings are negligible; Equal heat transfer area in each flashing stage; All physical properties of the seawater, brine, water vapor, and water condensate depend on temperature and salinity. The properties include the specific heat at constant pressure, dynamic viscosity, thermal conductivity, and density; The overall heat transfer coefficient is the sum of the thermal resistances expressed in terms of the inside and outside heat transfer coefficient, the fouling resistance, and the thermal resistance of the condenser tube; The latent heat of formed/condensed vapor depends on temperature; Thermodynamic losses include the boiling point elevation, the non-equilibrium allowance, and demister losses; The distillate product is salt free. Each flashing stage constitutes five balance equations; these are mass balance, salt balance, flashing brine energy balance, condenser energy balance, and condenser heat transfer rate. These relations are given by Eqns. 1-5.

$$B_{j-1} + \sum_{k=1}^{j-1} D_k = B_j + \sum_{k=1}^{j} D_k \tag{1}$$

$$X_{b_j} B_j = X_{j-1} B_{j-1} \tag{2}$$

$$D_j \lambda_{v_j} = B_{j-1} C_{pb} (T_{b_{j-1}} - T_{b_j}) \tag{3}$$

$$D_j \lambda_{c_j} + C_{pd} (T_{c_{j-1}} - T_{c_j}) \sum_{k=1}^{j-1} D_k = M_f C_{pf} (T_{f_j} - T_{f_{j+1}}) \tag{4}$$

$$M_f C_{pf} (T_{f_i} - T_{f_{i+1}}) = U_{c_j} A_c (LMTD)_{c_j} \tag{5}$$

In Eq. (5), the logarithmic mean temperature difference is given by

$$(LMTD)_{c_j} = (T_{f_i} - T_{f_{i+1}}) / \ln((T_{c_i} - T_{f_{i+1}}) / (T_{c_i} - T_{f_i})) \tag{6}$$

Therefore, the total number of equations is ($5n$), where (n) is the number of flashing stages. In addition, there is a system constraint, where the distillate mass flow is equal to the sum of the distillate product in all stages. This equation (constraint) is used to calculate the heat transfer area in the flashing stages (A), which is assumed equal in each stage.

3. Solution Algorithms

There are several approaches for solving the MSF model equations. The first is to use the root finding solvers, which includes methods like the conventional Newton-Raphson method, the globally convergent Newton's method, and the globally convergent Broyden's. A second approach is to use optimization solvers, which includes the nonlinear squares algorithm, the least-squares

algorithm, and the sequential quadratic algorithm [13]. The third class of solvers is the genetic algorithms (GA), which uses a stochastic solution approach. The GA implementation used in this work is the Augmented Lagrangian Genetic Algorithm (ALGA) [14]. The final solution algorithm is the Artificial Neural Network (ANN). In this work the back-propagation algorithm which is based on the general gradient descent method was implemented [15].

4. Results and Discussion

The MSF mechanistic model is coded in Matlab and is used to evaluate the residual functions needed by the solvers. A set of operating conditions and design parameters for a typical plant was selected as a basis for the model solution. Computation results for the root finding methods are summarized in Table (1).

Table (1) Solution results using the Numerical Methods

Solution Method	Objective Function	No. of Iterations	No. of Function Calls	CPU Time (sec)
Conventional Newton-Raphson	1.98e-04	200	24400	734.81
Global Convergent Newton	8.11e-06	7	859	74.31
Global Broyden's	7.03e-06	20	772	44.23

Results of the optimization based methods are shown in Table (2). The first method considered is the trust-region reflective method, which uses reflective Gauss-Newton. The CPU time and objective function values were reduced considerably when the same solver was reconfigured (Cases 1.2-1.7).

Table(2) Solution results using the Optimization Methods

No	Solution Method	Objective Function	No. of Iterations	No. of Func. Calls	CPU Time (sec)
1.1	Trust-region reflective Gauss-Newton	7.97E-05	166	20374	70.6
1.2	Gauss-Newton-BFGS, CubicPoly	1.25E-06	26	3173	10.7
1.3	Gauss-Newton-BFGS, QuadCubic	8.46E-09	32	4026	13.4
1.4	LM-BFGS, CubicPoly	1.21E-06	22	2685	8.93
1.5	LM-BFGS, QuadCubic	1.85E-09	30	3778	12.55
1.6	Dogleg-BFGS, CubicPoly	1.67E-09	18	2318	7.83
1.7	Dogleg-BFGS, QuadCubic	1.68E-09	18	2318	8.00
2	Nonlinear Least Squares	2.28E-04	30	3661	428.2
3	MiniMax	2.72E-04	46	5827	276.3
4	Constrained Minimization	2.06E-04	21	2553	75.90
5	Multi-objective Goal Attainment	Attain. Factor 0.4039	98	11275	691.8

Applying the GA algorithm to the MSF problem requires careful definition of the population diversity and size at each generation. In this problem the population size was selected same as the number of variables and the diversity was taken over a range of -1 to 10. Results for the GA algorithm are shown in Table (3).

Table (3) Solution results using the GA-based Methods

Case	Objective Function	No. of Iterations	No. of Func. Calls	CPU Time (sec)
(No GA)	7.97E-05	166	20374	70.61
Base Case	1.56E-06	33	4162	28.9
+20% of Base Case	5.88E-04	212	26500	81.5
- 20% of Base Case	2.45E-04	245	30380	104.1

Application of the ANN model requires sufficient data for training. Six input variables and twenty output variables were used to train a set of 6 ANNs. Each ANN receives the same input variables and predicts 3 output variables except the 6th ANN produces 4 outputs. The generated set of input output data covered most of the expected operational ranges of the variables. The results of training indicate that the training time for the first 5 networks was in the range 0.8-1.4 hrs whereas the last ANN took about 3.5 hr training time. Table (4) shows the simulation speed and prediction error for each of the 6 ANNs.

Table (4) Prediction performance of the ANN-based Method

ANN No.	Simulation Time (sec)	Relative Prediction Error
1	0.6623	1.64e-6
2	0.0099	4.60e-6
3	0.0098	4.95e-3
4	0.0093	4.58e-8
5	0.0102	3.70e-5
6	0.0098	8.72e-7

6. Conclusions

The MSF problem solution is so sensitive to the value of the initial guesses vector and most of the conventional techniques fail to converge when started away from the acceptable location of this vector. For such cases a priory prediction of the values for initial guesses using simple MSF models helps conventional solution algorithms to converge to a solution. Genetic algorithms can be used successfully for predicting reasonable initial guesses to be used later as inputs to conventional solvers. This way no matter what are the changes in the model, it can still be solved by conventional solvers. Comparison of the solution speeds for the tested solvers gives the following rank in descending

order: ANNs prediction (0.7113 s), Optimization using the Dogleg-BFGS, with Cubic Polynomial line search (7.83 s), GA-based optimization (28.9 s), and finally the global Broyden's solver (44.23 s). A reduction factor of about 11 is achieved between ANN-based solver and the fastest conventional solver. Hence, the ANN-based solvers were superior in reducing the computational effort. Other attractive features for using ANNs are the guaranteed accuracy of prediction and that it doesn't need initial guesses like other solvers. These features of the ANN-based predictors make them favorable than other conventional solution methods for applications involving model analysis, online parameter optimization and simulation applications.

Acknowledgement

The authors wish to acknowledge the financial support of the Middle East Desalination Research Center (MEDRC), Oman, project # 04-AS-001.

References

1. A.R. Glueck and W. Bradshaw, Proc., 3rd International Symposium on Fresh Water from Sea, 1 (1970) 95-108.
2. H. Hayakawa, H. Satori, and K. Konishi, Proc., 4th International Symposium on Fresh Water from the Sea, 1 (1973) 303-312.
3. A.M. Helal, M.S. Medani, M.A. Soliman, and J.R. Flower, Comp. Chem. Eng., 10 (1986) 327-342.
4. H. El-Dessouky and S. Bingulac, Methods Appl. Mech. Engrg. 141 (1997) 95-115.
5. B. A. Williams, Design of a Multistage Flash Desalination Plant by the Strong Maximum Principle, Thesis, Stanford University, Stanford, California, (1968).
6. J.H. Beamer and D.J. Wilde, Desalination, 9 (1971) 259-275.
7. M. Bourouis, L. Pibouleau, P. Floquet, and S. Domenech, Desalination 115 (1998) 1-14.
8. E.E. Tarifa E. E., D. Humana, S. Franco, and S.L. Martinez, Desalination 152 (2002) 215-222.
9. N. Srinivas and K. Deb, Evolutionary Computation, 2 (1995) 221-229.
10. R. Selvaraj and P.B. Deshpande, Desalination 101, 185-193 (1995).
11. K. A. Al-Shayji, Modeling, Simulation, and Optimization of Large-Scale Commercial Desalination Plants, Thesis, Virginia Polytechnic Institute and State University, (1998).
12. H. El-Dessouky and H. Ettouney, Fundamentals of salt water desalination, Elsevier, 2002.
13. T. Coleman, M.A. Branch, and A. Grace, Optimization Toolbox for use with MATLAB, User Guide, The Math Works Inc., Natick, MA, (1999).
14. A. R. Conn, N. I M. Gould, and P.L. Toint, Mathematics of Computation, 66 (1997) 261-288.
15. B. Müller and J. Reinhardt, Neural Networks: An Introduction. Springer-Verlag, New York, 1991.