A Genetic Hybrid for Critical Heat Flux Function Approximation

Yung-Keun Kwon
School of Computer Science
and Engineering
Seoul National University
Shilim-dong, Kwanak-gu
Seoul, 151-742 Korea
kwon@soar.snu.ac.kr

Sung-Deok Hong
Korea Atomic Energy Research Institute
Deokjin-dong, Yuseong-gu
Daejeon, 305-353 Korea
sdhong1@nanum.kaeri.re.kr

Byung-Ro Moon
School of Computer Science
and Engineering
Seoul National University
Shilim-dong, Kwanak-gu
Seoul, 151-742 Korea
moon@soar.snu.ac.kr

Abstract

Function approximation is the problem of finding a function that best explains the relationship between independent variables and a dependent variable. We propose a genetic hybrid for the critical heat flux function approximation which critically affects the performance of nuclear plants. The problem is represented for genetic algorithm in a way that exploits the relationships between parameters. The experimental result significantly improved the existing function at KAERI (Korea Atomic Energy Research Institute). The framework is not just for the tested problem; it is believed to be applicable to other function approximation problems.

1 Introduction

Given N data pairs $\{X_i, y_i\}$, $i = 1, 2, ..., N$, where each X_i is an n-dimensional vector of independent variables ($X_i = < x_{i_1}, x_{i_2}, \ldots, x_{i_n} >$) and y_i is a dependent variable, the function approximation problem (FAP) is finding a function that best explains the N pairs of X_i and y_i. Assume that the samples are derived from an underlying system of the following form:

$$y_i = f(X_i) + \triangle_i = f(x_{i_1}, x_{i_2}, \ldots, x_{i_n}) + \triangle_i.$$

A popular measure for the error with respect to a candidate function $\hat{f}$ is the LSE (Least Squares Error) which is defined as follows:

$$LSE(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(X_i))^2.$$

In a linear parametric model, we can find an optimal function by traditional regression analysis. When it is nonlinear, we cannot guarantee to find an optimal function in most cases. There have been a number of attempts to do function approximation with trainable dynamic systems using neural nets and fuzzy systems [8][30].

Derivative-based algorithm is a popular approach in a parametric model. However, every derivative-based algorithm converges to the nearest local minimum associated with the initial solution. Thus selecting a good starting point is critical for a derivative-based algorithm. A naive solution for this problem is the multi-start approach which applies a local optimization algorithm, such as a derivative-based algorithm, on a number of random starting points and returns the best result out of them. Another way is the Large-Step Markov Chain (LSMC) method which repeats a chain of "perturbation + local optimization" starting at an initial point. LSMC was popular in the 1990's, particularly for the traveling salesman problem [13][20]. Another is the hybrid genetic algorithms which showed notable successes on combinatorial optimization problems [5][6][21][23]. They generate diverse initial solutions by genetic operators and provide them as inputs for local optimization algorithms.

A hybrid of an adaptive regression splines algorithm and a genetic algorithm was used to solve some FAPs [25][26]. In the regression splines algorithm, the terms in a regression equation take the form of splines of the descriptors. If the numbers of descriptors and functional forms are small, the space of possible fitting equations can be explored exhaustively. However, if the numbers are large, this is not possible. In the hybrid, the search for function models was replaced by a genetic search. It showed better performance with shorter computation times. Another GA approach for designing a universal function approximator with a combination of trigonometric and polynomial basis functions was also proposed [1]. The result was reported to be better than that of a statistical regression based on polynomials, trigonometrics or cubic splines. It also outperformed a neural-network-based solution.

In this paper, we present a hybrid genetic algorithm for the critical-heat-flux (CHF) function approximation. It is a problem that critically affects the performance of nuclear plants. We use the nonlinear Levenberg-Marquardt algorithm for local optimization and combine it with a genetic search. In a genetic algorithm, it is known that the encoding of solutions significantly affects the performance [5]. We devise a parameter-reordering algorithm for genetic encoding to exploit the geographical relationships of parameters in the genetic search process.

This paper is organized as follows. In section 2, we explain the basics of Levenberg-Marquardt algorithm and critical heat flux, and present the objective. In section 3, we describe our approach for the critical-heat-flux function approximation. In section 4, we provide our experimental results and compare them against existing ones. Finally, conclusions are given in section 5.

2 Preliminaries

2.1 Levenberg-Marquardt algorithm

In linear systems, the steepest-descent algorithm, which moves in the steepest downhill direction determined by the gradient, is the basis for most derivative-based algorithms. Newton's method improves the steepest-descent algorithm by more efficiently determining the movement direction using a Hessian matrix, a matrix of the second partial derivatives. A major disadvantage of Newton's method is that calculating the inverse of the Hessian matrix is computationally expensive and may introduce numerical problems due to round-off errors. If the Hessian matrix is not positive definite, Newton's method may also move to a local maximum (saddle point) instead of a local minimum. Levenberg [18] and Marquardt [19] added a positive definite matrix to the Hessian matrix to make the Hessian positive definite. In this way, one can avoid being directed to a saddle point. This approach is generally called the Levenberg-Marquardt algorithm. For nonlinear systems, the starting point is the Gauss-Newton method which uses a Taylor series expansion to obtain a linear model that approximates the original nonlinear model. Then the least-square methods can be applied. Of course, the Levenberg-Marquardt algorithm can also be applied to this model; it is called the nonlinear Levenberg-Marquardt algorithm.

2.2 Critical Heat Flux

When a heated surface is wet with cooling liquid and most of the heat transferred is absorbed by the latent heat of vaporization, a large heat transfer can be achieved with a small temperature difference between the surface and liquid. However, the region of highly effective boiling heat transfer has a limiting boundary, and the limiting condition is called the *critical heat flux* condition.

The CHF condition is characterized by a sharp reduction of the local heat transfer coefficient which results from the replacement of liquid by vapor adjacent to the heat transfer surface. An occurrence of CHF is accompanied by an inordinate increase in the surface temperature for a surface-heat-flux-controlled system, and an inordinate decrease in the heat transfer rate for a surface-temperature-controlled system [27].

This can be explained with Newton's law of cooling as follows:

$$q = h(T_w - T_f)$$

where q, h, T_w, and T_f represent the heat flux, heat transfer coefficient, wall temperature, and fluid temperature, respectively. If h decreases significantly due to the occurrence of the CHF condition, T_w will increase for fixed q and T_f while q will decrease for fixed T_w and T_f.

The understanding of CHF phenomenon and accurate prediction of the CHF condition are important for safe and economic design of many heat transfer units including nuclear reactors, fossil-fuel boilers, fusion reactors, electronic chips, etc. Therefore, the phenomenon has been investigated extensively over the world since Nukiyama [24] first characterized it.

If CHF occurs in an atomic reactor, it lowers the thermal efficiency and hence causes a serious loss, which endangers the safety. Therefore, it is an important task to predict the occurrence of CHF under a certain condition. To find the CHF function, statistical techniques have been widely studied [2][10]; neural networks [22][31] and genetic programming [17] have also been tried.

2.3 The Objective

There can be a number of measures to evaluate the performance of an approximate function: the sum of the squared errors, the sum of the absolute errors, the maximum overshoot, etc. In our problem, we use *LRE* (Least Ratio Error) following the convention of the

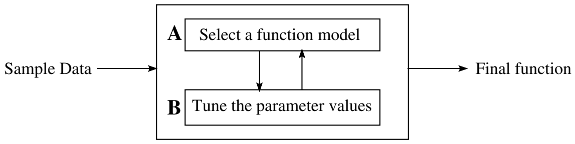

Figure 1: The process of function approximation

CHF studies [7][28]:

$$LRE(\hat{f}) = \frac{\sigma(\frac{y}{\hat{f}(X)})}{E(\frac{y}{\hat{f}(X)})}.$$

To clarify the meaning of the measure, we rewrite it as follows:

$$LRE(\hat{f}) = \sigma(\frac{y}{E(\frac{y}{\hat{f}(X)}) \times \hat{f}(X)}).$$

The final function that we evaluate is $E(\frac{y}{\hat{f}(X)}) \times \hat{f}(X)$ since the new function has the following useful property:

$$E(\frac{y}{E(\frac{y}{\hat{f}(X)}) \times \hat{f}(X)}) = 1.$$

2.4 The Dataset

Each data set consists of eight independent variables $x_1, \ldots, x_8$ and one dependent variable *CHF*. We were given 1607 sets of observed data from KAERI. The best known function with respect to *LRE* that KAERI has from years of tuning is as follows [14][15]:

$$\begin{aligned} CHF \quad = \quad & -0.019278x_1 - 0.17253\frac{x_2}{1000} \\ & - 0.1396tanh(0.05461(x_3 + x_4) - 1.97) \\ & - 0.38082x_5 - 0.054003x_6 \\ & - \{\frac{1.8987x_2}{1000} + \frac{0.047388x_1x_2}{1000} - 0.10821x_1 \\ & - 0.67613(\frac{x_2}{1000})^2\}x_7x_8 \\ & + 0.134698x_7 + 1.25103. \qquad (1) \end{aligned}$$

Since the actual meaning of the variables are beyond the focus of a methodological study, we renamed the original variables $x_1, \ldots, x_8$.

3 The Suggested GA

3.1 Our Approach

An ideal structure for an FAP is given in Figure 1. Given a sample data set, it repeats the process "i) select a function model, ii) tune the parameter values."

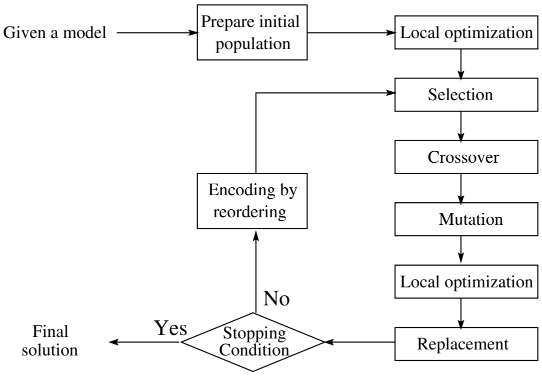

Figure 2: The structure of the RHGA

We may use a two-level genetic algorithm that finds both the function model and the coefficients by two genetic algorithms. In the scheme, the upper level GA provides function models and the other GA tunes the coefficients of each function model. This is an example of non-parametric optimization. The search space may be much wider than the GA can effectively solve in a practical time budget since both the function models and the sets of coefficients have unlimited numbers of eligible candidates. To cut the search space, we start with the best function model at KAERI mentioned in Section 2.4, and attempt to modify it by an analytical method. That is, the GA in this paper is used just for tuning the coefficients. We name this GA a Reordered Hybrid GA (RHGA).

The RHGA was applied to find coefficients in part B of Figure 1. Part A is tuned by an analytical method. The structure of the RHGA is shown in Figure 2. Given the function model (1) of Section 2.4, the coefficient distribution for training is as follows:

$$\begin{aligned} CHF \quad = \quad & a_1x_1 + a_2\frac{x_2}{1000} \\ & + a_3tanh(a_4(x_3 + x_4) + a_5) \\ & + a_6x_5 + a_7x_6 \\ & - \{\frac{a_8x_2}{1000} + \frac{a_9x_1x_2}{1000} + a_{10}x_1 \\ & + a_{11}(\frac{x_2}{1000})^2\}x_7x_8 \\ & + a_{12}x_7 + a_{13}. \qquad (2) \end{aligned}$$

The problem is to find the best set of coefficients a_1 through a_{13} with respect to the objective *LRE* in Section 2.3. In the following subsections, we describe each part of the RHGA in detail.

3.1.1 Problem Representation by Reordering

In the problem, the coefficients are all real numbers. Each solution is a set of 13 coefficient values. In a

GA, a solution is represented by a chromosome; here, a chromosome is a real array of 13 elements. Although binary representation has been popular in the GA community, real representation also has a long history dating back to the early 1960's [3][29]. Each element of the array is called a gene and we restrict the range of each gene to [-50, 50].

3.1.2 Operations: Selection, Crossover, Mutation

Two parent chromosomes are selected with probabilities that are proportional to their fitness values. The fitness values are normalized in such a way that the best chromosome is chosen with a probability four times higher than that of the worst chromosome. This is a general practice in the GA community [11]. The normalized fitness value of a chromosome in the population is computed as follows:

$$F_k = Q_w - Q_k + (Q_w - Q_b)/3,$$
$$Q_k = \sigma(\frac{y}{\hat{f}_k(X)})/E(\frac{y}{\hat{f}_k(X)})$$

where

F_k : fitness of chromosome k
$\hat{f}_k$: the function corresponding to chromosome k
b, w : the indices of the best and the worst chromosomes in the population

A crossover operator creates a new offspring chromosome by combining parts of the two parent chromosomes. RHGA uses 3-point crossover that works as follows. It randomly selects three cut points in the same positions on both parent chromosomes. The cut points divide each chromosome into four disjoint parts. It makes an offspring by alternately copying the parts from the two parents. RHGA then perturbs the solution with the following mutation operator. It generates a random number for each gene of the offspring. If the random number for the gene is smaller than a preset probability P_1, it is replaced with an arbitrary number in the range [-50,50].

3.1.3 Local Optimization

Local optimization is performed on each offspring after crossover and mutation. Generally a GA is inefficient in fine-tuning around local optima. A local optimization algorithm helps a GA fine-tune and improves its convergence. RHGA uses the nonlinear Levenberg-Marquardt algorithm for local optimization. The Levenberg-Marquardt algorithm takes a set of initial coefficients as input, and outputs a locally optimized set of coefficients. The GA provides diverse initial solutions by crossover and mutation for the Levenberg-Marquardt algorithm.

3.1.4 Replacement Operation and Stopping Criterion

RHGA uses the replacement operator used in [5]. The offspring first attempts to replace the parent more similar to itself, measured by the sum of the distances between all coefficient pairs. If it fails, it attempts to replace the other parent (replacement is done only when the offspring is better than one of the parents). If the offspring is worse than both parents, it replaces the most inferior member of the population. It stops after a given number of generations.

3.2 Reordering and the Modification of Function Models

3.2.1 Coefficient Reordering

A *schema* is a pattern inside chromosomes. Given a set of alphabets S, a schema is defined to be an n-tuple $s_1 s_2 \ldots s_n$ where $s_i \in S \cup \{*\}$. In a schema, the symbol "$*$" specifies the don't-care positions and the other symbols are *specific symbols* which specify the pattern. The *defining length* of a schema is defined to be the length from the leftmost specific symbol to the rightmost specific symbol. We call a schema with k specific symbols a k^{th}*-order* schema. Some schemas survive and some do not by a crossover operator. The survival of high-quality schemas is important since GAs can be explained as a growing process from low-order schemata to high-order schemata [12]. In a single-point crossover, schemas with short defining lengths have higher probabilities to survive over generations. If we use multipoint crossovers, a schema is not disrupted when an even number of crossover points fall between the two specific symbols of every pair of adjacent specific symbols. The survival probability of a schema is not only affected by its defining length and we have to consider the distribution of specific symbols [6]. For example, consider two 6-order schemas H_1 and H_2 with the same defining length of 20. Specific symbols are evenly distributed in H_1 but they are highly clustered in H_2. When two-point crossover is used, the survival probability of H_1 is 45/325, and that of H_2 is 120/325. H_2 has a much higher probability of survival.

H_1 : ***#***#***#***#***#***#***
H_2 : ***###***************###***

This example shows the importance of genes' geographical distribution in the chromosomal representation of a GA. If two genes have a strong relationship, it is advantageous to locate them closely [4][5]; in this problem, we suggest a reordering algorithm that uses

```
Calculate $Corr(c_i, c_j)(i, j = 1, 2, \ldots, L)$;
Find the pair $(c_m, c_n)(m \neq n)$ having the highest correlation;
$S = c_m c_n$;
$U = \{c_1, c_2, \ldots, c_L\} - \{c_m, c_n\}$;
while ( $U \neq \emptyset$ )
{
        Find $c_l$ having the highest value $F_l(c_l, S)$;
        Find $c_r$ having the highest value $F_r(S, c_r)$;
        if ( $F_l(c_l, S) > F_r(S, c_r)$ ) {
                $S = c_l \cdot S$; //concatenation
                $U = U - \{c_l\}$;
        } else {
                $S = S \cdot c_r$; //concatenation
                $U = U - \{c_r\}$;
        }
}
```

Figure 3: Reordering algorithm

the correlations between all the pairs of coefficients. Figure 3 shows the coefficient-reordering algorithm. In the algorithm, functions F_l and F_r compute the correlation between a coefficient c and a string, S, of coefficients as follows:

$$F_l(c, S) = \alpha \times Corr(c, c_1) + (1 - \alpha) \times Corr(c, c_2)$$
$$F_r(S, c) = \alpha \times Corr(c_k, c) + (1 - \alpha) \times Corr(c_{k-1}, c),$$

where

$$S = c_1 c_2 \ldots c_k$$
$$Corr(a, b) = \frac{E[(a - \bar{a})(b - \bar{b})]}{\sigma_a \sigma_b}$$
$$\alpha : \text{a weight.}$$

In computing the correlation between a coefficient and a string S, it only considers the two leftmost or rightmost coefficients in the string S. The reasonable range for α is [0.5, 1]. If α=1, only the leftmost or rightmost coefficient is considered. The main purpose of the reordering is to reduce the probability that a crossover operator separates coefficients with high correlations. The reordering helps the pairs of coefficients having high correlations to stay close in chromosomes.

3.2.2 Modification of Function Models

Although we do not intend non-parametric optimization, we attempt to modify the function model (2) of page 3. We examine whether each term of the function properly explains the data with the solution obtained by RHGA. We modify the function model according to that examination. Formally, we transform the function with respect to a coefficient x_k as follows:

$$y = G(x_1, x_2, \ldots, x_n)$$
$$\Longleftrightarrow G_1(x_k) = G_2(y, x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n).$$

We plot the relationship between $G_2(y,\ x_1,\ \ldots,\ x_{k-1},\ x_{k+1},\ \ldots,\ x_n\ )$ and x_k, and also plot another relationship between $G_1(x_k)$ and x_k. If those two relationships are visibly different from each other, it is considered a signal to modify the function model with respect to the variable x_k. Not all variables are capable of being examined in this way; the variables x_3, x_4, x_5, x_6 and x_8 are those so capable.

The function model	KAERI	RHGA
$E(y/\hat{f}(X))$	1.0026714	0.9982438
$\sigma(y/\hat{f}(X))$	0.1072649	0.0996506
LRE	0.1069791	0.0998259

Table 1: Quality of KAERI and RHGA model

We attempted to modify the function model in this way and by adding some linear terms. We have observed that G_1 and G_2 are inconsistent with respect to x_4, x_5, and x_6. We modified the terms relevant to them as follows:

$$\begin{aligned} CHF \quad = \quad & a_1 x_1 + a_2 \frac{x_2}{1000} \\ & + a_3 tanh(a_4 x_3 + a_5 log(x_4) + a_6) \\ & + a_7 x_5 + a_8 x_5^{a_9} + a_{10} x_6 + a_{11} x_6^{a_{12}} \\ & - \{\frac{a_{13} x_2}{1000} + \frac{a_{14} x_1 x_2}{1000} + a_{15} x_1 + \\ & + a_{16} (\frac{x_2}{1000})^2\} x_7 x_8 \\ & + a_{17} x_7 + a_{18} + a_{19} \frac{x_8}{100}. \end{aligned} \tag{3}$$

4 Experimental Results

For robust comparison between the KAERI model and the RHGA model we follow the 10-fold cross-validation approach [9][16]. We randomly split the entire dataset D into 10 mutually exclusive subsets $D_1, D_2, \ldots, D_{10}$ of approximately equal size. The RHGA is trained and tested 10 times; the k^{th} experiment was trained with $D \setminus D_k$ and tested with D_k.

The cross-validation estimate of the average and the standard deviations of the observed CHF value over the predicted value are shown in Table 1. In the table, LRE, described in section 2.3, is the most popular measure for errors in the CHF approximation in the nuclear engineering community. The RHGA approach outperformed the KAERI function by about 7%.

	# of Generations	*LRE*	std-dev	trials
RHGA without reordering	7980.20	0.1006804	0.0005521	50
RHGA with reordering	5105.69	0.1001852	0.0002063	50

Table 2: The Effect of Reordering

Table 2 shows the effect of reordering. In the table, "# of Generations" represents the average generation in which the best solution has appeared. The reordering improved the solution quality in visibly less time.

5 Conclusions

In this paper, we proposed a genetic algorithm for the CHF function approximation problem that combines the genetic search with a nonlinear Levenberg-Marquardt algorithm. The Levenberg-Marquardt algorithm helps the GA to fine-tune, and the GA helps the Levenberg-Marquardt algorithm to overcome its narrow scope. We also proposed a coefficient-reordering algorithm to exploit the geographical relationships of genes in the genetic encoding, which also turned out to contribute to the performance improvement. We should note that the function models were not decided by a search method (e.g., a genetic algorithm) but by analytic modification. It may be worth giving more freedom to the forms of function models under a fully non-parametric optimization model. There is a trade-off. We are sure that giving *full* freedom is not the right approach unless the computing power is strengthened by exponential orders of magnitude. Our current result is 7% better than the best known solution.

Acknowledgement

This work was supported by Statistical Research Center for Complex Systems and Brain Korea 21 Project at Seoul National University. The RIACT at Seoul National University provided research facilities for this study.

References

[1] M. A. Ahmed and K. A. De Jong. Function approximator design using genetic algorithm. In *IEEE Conf. on Evolutionary Computation*, pages 519–524, 1997.

[2] R. W. Bowring. A simple but accurate round tube, uniform heat flux, dryout correlation over the pressure range 0.7-17 mn/m. In *AEEW-R*, page 789, 1972.

[3] H. Bremermann. Optimization through evolution and recombination. *Self-Organizing Systems*, 1962.

[4] T. N. Bui and B. R. Moon. Hyperplane synthesis for genetic algorithms. In *International Conference on Genetic Algorithms*, pages 102–109, 1993.

[5] T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Trans. on Computers*, 45(7):841–855, 1996.

[6] T. N. Bui and B. R. Moon. GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 17(3):193–204, 1998.

[7] G. P. Celata, M. Cumo, Y. Katto, and A. Mariani. Prediction of the critical heat flux in water subcooled flow boiling using a new mechanistic approach. *International Journal of Heat and Mass Transfer*, 42:1457–1466, 1999.

[8] J. A. Dickerson and B. Kosko. Fuzzy function approximation with ellipsoidal rules. *IEEE Trans. on Systems, Man and Cybernetics*, 26(4):542–560, 1996.

[9] B. Efron and R. Tibshirani. *Cross-validation and the bootstrap: Estimating the error rate of a prediction rule.* Technical Report (TR-477), Dept. of Statistics, Stanford University., 1995.

[10] Y. Fujita. Prediction methods of heat transfer coefficient and critical heat flux in mixture boiling. *Experimental Heat Transfer, Fluid Mechanics and Thermodynamics*, pages 831–842, 1997.

[11] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[12] J. Holland. *Adaptation in Natural and Artificial Systems.* Univ. of Michigan Press, 1975.

[13] I. Hong, A. B. Kahng, and B. R. Moon. Improved large-step markov chain variants for the symmetric TSP. *Journal of Heuristics*, 3(1):63–81, 1997.

[14] D. H. Hwang. Study on the critical heat flux. KAERI/AR-339/91, Korea Atomic Energy Research Institute.

[15] D. H. Hwang, Y. J. Yoo, J. R. Park, and Y. J. Kim. Evaluation of the thermal margin in a KOFA-loaded core by a multichannel analysis methodology. *Journal of Korea Nuclear Society*, 27:518–531, 1995.

[16] R. Kohavi. A study of cross-validation and bootstrap of accuracy estimation and model selection. In *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[17] D. G. Lee. Critical heat flux prediction using genetic programming for water flow in vertical round tubes. *International Comm. Heat Mass Transfer*, 24:919–929, 1997.

[18] K. Levenberg. A method for the solution of certain problems in least squares. *Applied Mathematics*, 2:164–168, 1944.

[19] D. W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 1963.

[20] O. Martin, S. W. Otto, and E. W. Felten. Large-step markov chains for the traveling salesman problem. *Complex Systems*, 5(3):299–326, 1991.

[21] P. Merz and B. Freisleben. Genetic local search for the TSP:New results. In *IEEE Conf. on Evolutionary Computation*, pages 159–164, 1997.

[22] S. K. Moon, W. P. Baek, and S. H. Chang. Parametric trends analysis of the critical heat flux based on artificial neural networks. *Nuclear Engineering and Design*, 12:29–49, 1996.

[23] Y. Nagata and S. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the travelling salesman problem. In *International Conference on Genetic Algorithms*, pages 450–457, 1997.

[24] S. Nukiyama. The maximum values of the heat Q transmitted from metal to boiling water under atmospheric pressure. *Trans: JSME*, 37:357, 1934.

[25] D. Rogers. G/splines: A hybrid of friedman's multivariate adaptive regression splines (mars) algorithm with holland's genetic algorithm. In *International Conference on Genetic Algorithms*, pages 384–391, 1991.

[26] D. Rogers. Evolutionary statics: Using a genetic algorithm and model reduction to isolate alternate statistical hypotheses of experimental data. In *International Conference on Genetic Algorithms*, pages 735–742, 1997.

[27] T. G. Theofanous. The boiling crisis in nuclear reactor safety and perfomance. *International Journal of Multiphase Flow*, 10:69–95, 1980.

[28] K. C. Tu, C. H. Lee, S.J. Wang, and B. S. Pei. A new mechanistic critical heat flux model at low-pressure and low-flow conditions. *Nuclear technology*, 124:243–254, 1998.

[29] R. Weinberg. *Computer Simulation of a Living Cell*. Ph.D. thesis, University of Michigan, 1970.

[30] S. S. Yang and C. S. Tseng. An orthogonal neural network for function approximation. *IEEE Trans. on Systems, Man and Cybernetics*, 26(5):779–785, 1996.

[31] T. C. Yapo, M. J. Embrechts, S. T. Cathey, and R. T. Lahey. Prediction of critical heat fluxes using a hybrid kohonen-backpropagation neural network. *Intelligent Engineering Systems through Artificial Neural Networks*, 2:853–858, 1992.