Available online at www.sciencedirect.com
ScienceDirect
International Journal of Heat and Mass Transfer 51 (2008) 2299–2312
International Journal of HEAT and MASS TRANSFER
www.elsevier.com/locate/ijhmt

Optimization of the location of multiple discrete heat sources in a ventilated cavity using artificial neural networks and micro genetic algorithm

Rajeev Reddy Madadi, C. Balaji *

Heat Transfer and Thermal Power Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai – 600036, Tamil Nadu, India

Received 21 May 2007; received in revised form 27 July 2007
Available online 29 October 2007

Abstract

This work is aimed at evaluating the optimal location of three discrete heat sources which could be placed anywhere inside a ventilated cavity and cooled by forced convection. The computational domain involves a square cavity with adiabatic walls, diagonally opposite inlet and outlet, with a heat flux of 1000 W/m^2 on the heat sources and constant velocity of 4 m/s at the inlet. The two dimensional flow and temperature fields are obtained by performing simulations on FLUENT 6.3. The micro genetic algorithm (MGA) using the six coordinates of the heat sources as input parameters and 5 individuals in a population is used for the optimization, with the objective function as minimizing the maximum temperature on any of the heat sources. Initially for 66 generations, simulations were repeatedly done to evaluate the objective function. This data was used to train a back-propagation artificial neural network (ANN) using the Bayesian regularization algorithm to predict the fitness from the six inputs. This trained ANN was integrated with the micro genetic algorithm to evolve the population for 1000 generations to arrive at the global optimum. Sensitivity studies have been carried out on the optimal solution by varying the Reynolds number. This study shows that by integrating ANN with GA, the computational time can be reduced substantially in problems of this class.

Keywords: Optimization; Discrete heat sources; Constructal; Turbulent flow; Forced Convection; Micro genetic algorithm (MGA); Artificial neural networks

1. Introduction

Electronic equipment have become an integral part of almost every phase of modern living. Better reliability of this equipment requires that, they be maintained at a relatively constant temperature that should be below the maximum service temperature specified by the manufacturer. Consequent to the miniaturization of electronic equipment, the heat fluxes increase which in turn demands more efficient cooling strategies. While liquid cooling may provide the answer to this, air cooling is still prevalent up to certain heat flux levels due to its simplicity and low cost. It is also very important that these cooling systems are designed in the most efficient way. It is in this context that optimization studies of the location of the heat sources are becoming increasingly relevant.

Forced convection in electronics cooling has attracted considerable attention in the literature. Icoz et al. [1] have analyzed the cooling of two discrete heat sources placed in a horizontal channel. They showed that the heat transfer rates increase by 35–70% in turbulent regime and pressure drop, which is a more severe constraint, increases with the height of the heat source and the Reynolds number. Bhowmik et al. [2] have studied mixed, natural and forced

* Corresponding author. Tel.: +91 044 2257 4689; fax: +91 044 2257 0509.
E-mail address: balaji@iitm.ac.in (C. Balaji).

0017-9310/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2007.08.033

Nomenclature

a_i	neural network predicted value
b	height of the cavity (m)
$C_{1\varepsilon}$, $C_{2\varepsilon}$, $C_{3\varepsilon}$	$k-\varepsilon$ model constants
d	width of the inlet to the cavity (m)
$E_D(w)$	weight decay regularizer
$E_w(w)$	sum squared error between predicted and test data
$F(w)$	performance function for Bayesian regularization
I	turbulence intensity $\frac{u'}{u_{avg}}$
k	turbulence kinetic energy ($m^2\ s^{-2}$)
L	width of the cavity (m)
L_h	length of the heat source (m)
M	number of nodes in the x-direction
MRE	absolute mean relative error
N	number of nodes in the y-direction
N_{pop}	number of individuals in a population
P	static pressure (Pa)
Pr	Prandtl Number $\left(\frac{\mu C_p}{k}\right)$
Δp_{max}	maximum pressure drop inside the cavity (Pa)
q'	heat flux on the heat source ($W\ m^{-2}$)
Re	Reynolds Number $\left(\frac{\rho u d}{\mu}\right)$
R^2	absolute fraction of variance
R_{test}	correlation coefficient on test data set
T_{max}	maximum temperature on the heat sources (K)
T_∞	temperature of inlet air (K)
t_i	available target data for ANN
t_h	thickness of the heat source (m)
u	velocity component of fluid in the x-direction ($m\ s^{-1}$)
u_∞	velocity at the inlet of the cavity ($m\ s^{-1}$)
u_τ	friction velocity $\left(\sqrt{\tau_w/\rho}\right)$ near the wall ($m\ s^{-1}$)
v	velocity component of fluid in the y-direction ($m\ s^{-1}$)
w	network weight
x_i	horizontal coordinate of mid point of ith heat source (m)
y_i	vertical coordinate of mid point of ith heat source (m)
y^+	dimensionless wall distance $\left(\frac{\rho u_\tau y_n}{\mu}\right)$
y_n	normal distance of the cell centre from nearest wall (m)

Greek symbols

α	fraction of sum square error in Bayesian algorithm
β	fraction of weight decay regularizer used in Bayesian algorithm
β'	coefficient of thermal expansion (K^{-1})
ε	turbulence dissipation rate ($m^2\ s^{-3}$)
λ	number of bits used to encode the input variables to the micro genetic algorithm
μ	dynamic viscosity ($kg\ m^{-1}\ s^{-1}$)
μ_t	turbulent viscosity ($kg\ m^{-1}\ s^{-1}$)
ρ	density of the fluid ($kg\ m^{-3}$)
σ_k, σ_T, σ_ε	turbulent Prandtl numbers for k, T and ε
τ_w	wall shear stress ($N\ m^{-2}$)

convection heat transfer of four inline chips in a vertical channel. The effects of heat flux, coolant flow rates and chip numbers were investigated and a correlation was developed to describe the three modes of convection. Mathews and Balaji [3] have carried out a numerical investigation of turbulent mixed convection in a vertical channel with discrete heat sources and studied the effect of various parameters like the thermal conductivity of wall material, thermal conductivity of discrete heat source (k_c), Reynolds number and so on. Using the ACFD approach, a correlation to evaluate the maximum temperature on the heat sources on the channel has been developed which suits the engineer's interest. Choi and Ortega [4] have studied mixed convection in an inclined channel with discrete heat sources. The results show that the overall Nusselt number strongly depends on the inclination angle in the natural and mixed convection regime and is insensitive in the forced convection regime.

Constructal theory was used to study the optimal distribution of discrete heat sources mounted on a horizontal wall with forced convection by da Silva et al. [5]. With the objective to increase the global conductance between the hot wall and coolant, their results showed that the heat sources have to be distributed non-uniformly. Simulations of the flow regime have shown that as the Reynolds number increases, the heat sources tend to migrate towards the tip of the boundary layer. For a fixed heater area, the cooling increases with the number of optimally placed heat sources. Recently, da Silva et al. [6] have determined the optimal location of heat sources on a vertical wall with natural convection using constructal theory. The results showed that the optimal locations are a strong function of the Rayleigh number and the heat sources move towards the tip of the boundary layer at higher Rayleigh numbers. An interesting feature of this work is that as the number of heat sources increases, the rate of increase of global conductance decreases which shows that diminishing returns is an important characteristic of optimized complex structures. Dias and Milanez [7] have proposed a new methodology in the optimization studies of heat sources on a vertical wall. They have validated the results of da Silva et al. [6] by using genetic algorithms and showed that this approach is computationally more feasible and faster than an extensive search technique.

Based on the above literature survey, it can be concluded that most of the research has been focused on heat sources mounted on a horizontal, vertical or an inclined channel. The idea of moving the heat sources anywhere in a confined flow is also closely related to constructal theory which has been reviewed extensively in [8]. This geometry of heat sources placed anywhere inside a cavity is the basic building block for studying more complex situations like avionic packages, computer cabins and so on. Hence, in this paper we attempt a contemporary approach of combining artificial neural networks with genetic algorithm to determine a global optimum for the problem.

2. Model and governing equations

2.1. Model

The geometry for the problem under consideration is shown in Fig. 1. The two dimensional cavity shown has dimensions of $L = 0.4$ m and $b = 0.4$ m and is ventilated where air flows in from the bottom left corner and flows out from the top right corner. There are three heat sources that need to be cooled by the aid of a fan and these, in principle, can be anywhere inside the cavity. Each heat source is modeled as a line heat source with a height $L_h = 0.04$ m and thickness $t_h = 8 \times 10^{-3}$ m, with a constant heat flux of 1000 W/m^2 on two sides. The fluid enters the inlet with a constant velocity of $u_\infty = 4$ m/s and the Reynolds number $Re = 21797$ which corresponds to a highly turbulent regime. The turbulent intensity at the inlet is chosen as 10%. The flow exit is modeled as a pressure outlet. All the outer walls are assumed to be adiabatic.

2.2. Basic equations

The turbulent flow of air is described by the Reynolds Averaged Navier Stokes (RANS) equations. These equations for a two dimensional, steady, incompressible flow and heat transfer are as follows:

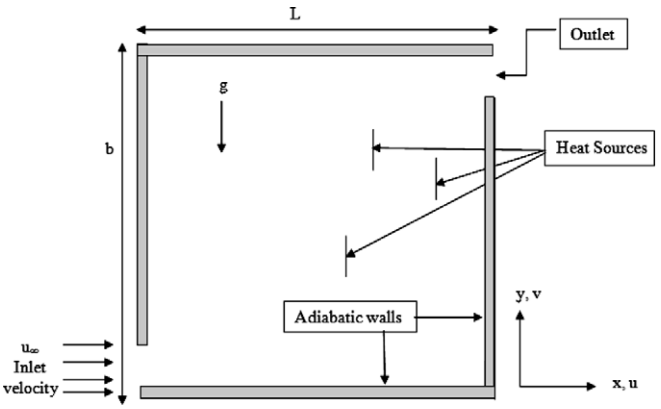

Fig. 1. Schematic of the problem geometry.

Continuity equation:

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{1}$$

Momentum equations:

$$\frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho v u) = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left[2(\mu + \mu_t)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[(\mu + \mu_t)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] \tag{2}$$

$$\frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v v) = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\left[(\mu + \mu_t)\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial y}\left[2(\mu + \mu_t)\frac{\partial v}{\partial y}\right] \tag{3}$$

Energy equation:

$$\frac{\partial}{\partial x}(\rho u T) + \frac{\partial}{\partial y}(\rho v T) = \frac{\partial}{\partial x}\left[\left(\frac{\mu}{Pr} + \frac{\mu_t}{\sigma_T}\right)\frac{\partial T}{\partial x}\right] + \frac{\partial}{\partial y}\left[\left(\frac{\mu}{Pr} + \frac{\mu_t}{\sigma_T}\right)\frac{\partial T}{\partial y}\right] \tag{4}$$

The standard $k-\varepsilon$ turbulence model has been the workhorse of practical engineering flow calculations due to its robustness, economy and reasonable accuracy for a wide range of turbulent flows. Hence in this study, the standard $k-\varepsilon$ model is used. The turbulence kinetic energy (k) and turbulent dissipation rate (ε) are evaluated at every point in the flow from a solution to the following transport equations.

$$\frac{\partial}{\partial x}(\rho k u) + \frac{\partial}{\partial y}(\rho k v) = \frac{\partial}{\partial x}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x}\right] + \frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\right]\frac{\partial k}{\partial y} + G_k + P_k - \rho\varepsilon \tag{5}$$

$$\frac{\partial}{\partial x}(\rho \varepsilon u) + \frac{\partial}{\partial y}(\rho \varepsilon v) = \frac{\partial}{\partial x}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x}\right] + \frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\right]\frac{\partial \varepsilon}{\partial y} + C_{1\varepsilon}\frac{\varepsilon}{k}(P_k + C_{3\varepsilon}G_k) - C_{2\varepsilon}\frac{\varepsilon^2}{k} \tag{6}$$

where

$$P_k = \mu_t\left[2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2\right] \tag{7}$$

$$G_k = -\frac{\mu_t}{\sigma_T} g \beta' \frac{\partial T}{\partial y} \tag{8}$$

The turbulence viscosity μ_t is evaluated from the velocity scale ($k^{1/2}$) and the length scale $\left(\frac{k^{3/2}}{\varepsilon}\right)$ and is defined in Eq. (9).

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{9}$$

The constants used in the above equations are given by $C_{1\varepsilon} = 1.44$, $C_{2\varepsilon} = 1.92$, $C_\mu = 0.09$, $\sigma_k = 1$, $\sigma_\varepsilon = 1.3$

$$C_{3\varepsilon} = \tanh\left|\frac{v}{u}\right| \tag{10}$$

3. CFD solution procedure

3.1. Numerical scheme

The two dimensional governing equations are discretized on a non-uniform triangular grid using the finite volume method. Since the heat sources are in the y-direction, the grid was created with the number of nodes in the y-direction being twice the number of nodes in the x-direction to capture the temperature field at the heat sources accurately. The velocities and the pressure are calculated using the semi implicit pressure linked equation solver (SIMPLE) algorithm. The interpolation of the gradients of velocities and temperature used the second order upwind scheme. The discretized equations are then linearized using an implicit technique and solved iteratively using FLUENT 6.3 [9]. As the standard $k-\varepsilon$ model is only valid for turbulent core flows, enhanced wall treatment is employed to use the $k-\varepsilon$ model in the near wall region. For this purpose, the mesh near the walls is sufficiently refined and the y^+ values near the wall vary between 0 and 0.92 which indicates the adequacy of the enhanced wall treatment approach used in the present study.

3.2. Solution verification

Solution verification deals with the assessment of the numerical errors which always exist when partial differential equations are solved numerically. The discretization error, which is the difference between the numerical solution and the exact solution to the continuum partial differential equation, is instrumental in establishing the accuracy of the CFD solution. Richardson extrapolation (Christopher J. Roy [10]) is a popular approach for obtaining a grid which has low discretization error. It requires numerical solutions on two or more grids with different levels of refinement which are then used to obtain a higher-order estimate of the exact solution.

As shown in Table 1, simulations were done on different grid sizes with the number of nodes varying from 15,994 to 63,470 for the grid independence study. It is also clear that as the grid is refined by increasing the number of nodes, the maximum temperature progressively increases. Fig. 2 shows a plot of the maximum temperature vs. $1/M \times N$ where M is the number of nodes in the x-direction and N is the number of nodes in the y-direction. A best fit line is also drawn whose intercept on y-axis corresponds to the temperature on a grid of infinite nodes. This grid independent temperature $T_{\inf}$ is found out to be 354.1 K. The relative error of temperature on grid 9 of 352.36 K with $T_{\inf}$ is 0.49 %. Hence grid 9 with 25,000 nodes is chosen for further analysis. Table 2 shows the deviation between the energy and mass inflow and outflow. The negligible deviation between the two quantities is a further indication of the accuracy of the solution. To ascertain the insensitivity of the results to the turbulence intensity (I), studies have been done on a typical case with the turbulence intensity changing from 5% to 25% which resulted in less than 0.4% change on the maximum temperature on the heat sources.

Table 1
Variation of temperature with grid size for the computational domain employed in the present study

S. No.	Grid size (M × N)	Total nodes (n)	Maximum temperature (K)
1	25 × 50	15,994	347.91
2	30 × 60	11,730	348.12
3	35 × 70	14,347	348.53
4	40 × 80	17,069	349.36
5	45 × 90	20,085	349.39
6	50 × 100	23,244	349.82
7	55 × 110	19,128	351.25
8	60 × 120	22,009	350.90
9	65 × 130	25,042	352.36
10	100 × 200	51,765	352.68
11	120 × 240	63,470	353.21

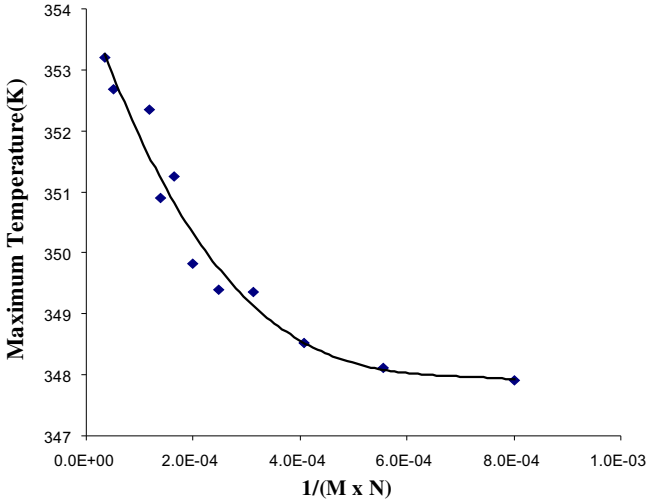

Fig. 2. Grid independence and Richardson extrapolation.

Table 2
Energy and mass balance of the problem analyzed in the present study showing the accuracy of the solution for a velocity of 4 m/s and heat flux of 1000 W/m^2

	Value (in)	Value (out)	% Deviation
Energy (kW)	3.73314	3.73315	2.41E−05
Mass flow (kg/s)	1.715	1.715001	5.83E−05

3.3. Solution validation

The turbulent modeling approach employed in the present study is validated with the experimental data of turbulent natural convection in a square cavity provided by Ampofo and Karayiannis [11]. The isothermal left hot wall

and right cold wall were maintained at 50 °C and 10 °C, respectively. Though in [11], exact adiabaticity could not be achieved on the horizontal walls, the comparison should serve as a ballpark indicator of the adequacy of the numerical model. The vertical velocity and the temperature of the fluid were evaluated along a horizontal line drawn at mid-height in the square cavity and the results plotted in Figs. 3 and 4.

Fig. 3 shows that vertical velocity at the wall is zero and increases rapidly very close to the wall and finally becomes zero in the centre region of the cavity. Similarly due to the thermal boundary layer developed near the walls, the temperature increases near the hot wall and decreases near the cold wall. Even though, the top and bottom wall are not truly adiabatic in the experimental study in [11], the velocity and temperature profiles at mid-height are similar to the ideal case of adiabatic walls. Therefore, the excellent correlation between the data available from [11] and the present study "qualitatively" validates the methodology adopted in this study.

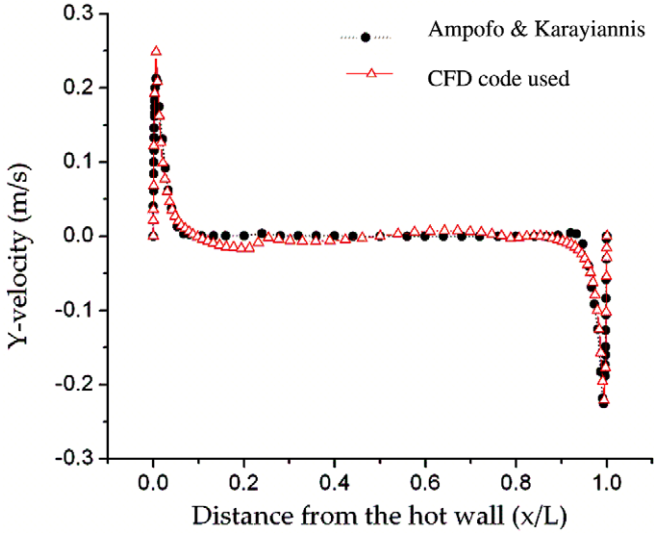

Fig. 3. Validation vertical velocity profile at $(y/L) = 0.5$ for the benchmark problem.

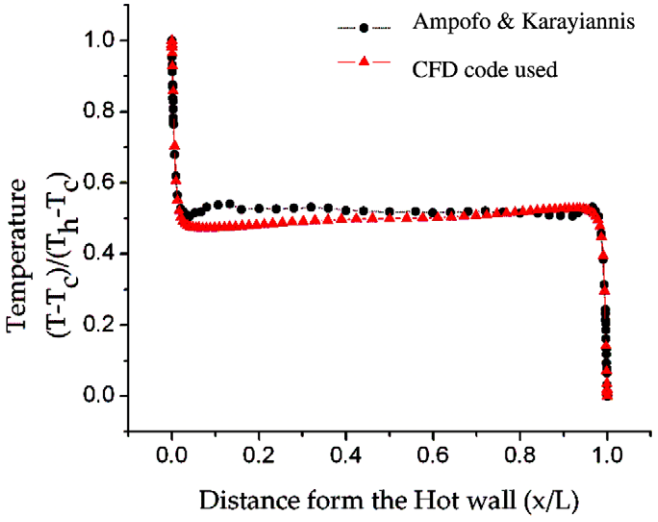

Fig. 4. Validation temperature profile at $(y/L) = 0.5$ for the benchmark problem.

4. Optimization with genetic algorithm

4.1. Genetic algorithm

Optimization plays an important role in today's world and is especially critical for design engineers when multiple parameters are involved. There are several techniques for optimization like analytical approach, downhill simplex method (Nelder and Mead [12]), gradient descent and so on. However most of the above algorithms suffer from the disadvantage of getting "stuck" in the local minimum, and so recently evolutionary algorithms like genetic algorithms (Holland [13], Goldberg [14]), simulated annealing (Kirkpatrick et al. [15]) and ant colony optimization (Dorigo and Maria [16]) have been proposed to overcome these difficulties of the traditional algorithms.

The genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that minimizes the "cost function". GA has remarkable abilities which include being able to solve non-smooth, non-continuous, non-differentiable cost functions, to escape the local optima and to move towards the global optimum.

Due to these inherent advantages, the genetic algorithm (GA) is employed in this study as the solution space is vast and several local optima should be possible. As the heat sources are of constant length, the coordinates of their mid-points are taken as the input variables to GA. Hence there are six input variables x_1, y_1, x_2, y_2, x_3, y_3 corresponding to the mid-points of the three heat sources, respectively. The cost function (C) used in the GA is defined as follows:

$$C = -\left[\frac{q''}{k(T_{\max} - T_{\infty})}\right] \tag{11}$$

The objective of the optimization is to minimize the cost function (C) which would lead to the minimization of $T_{\max}$. To calculate the population fitness values, a numerical simulation must be performed, until convergence, with the corresponding set of heat sources locations for each individual. Since several individuals and populations must be evaluated, the computational cost may become prohibitive. Due to this, the micro genetic algorithm (MGA) (Krishnakumar [17]) is used, since it requires a reduced population size, and therefore fewer fitness function evaluations.

4.2. Micro genetic algorithm

Micro genetic algorithms (Krishnakumar [17]) evolve very small populations that are very efficient in locating

promising areas of the search space. As the small populations are unable to maintain diversity for many generations, the population is restarted whenever diversity is lost, keeping only the very best fit individuals. Restarting the population several times during the run of the genetic algorithm has the added benefit of preventing premature convergence due to the presence of a particularly fit individual, which poses the risk of preventing further exploration of the search space and so may make the program converge to a local minimum.

Fig. 5 shows the flow chart for implementing the micro genetic algorithm. Micro GA randomly initializes N_{pop}, string individuals. For every generation (iteration), MGA performs six basic operations: variable encoding and decoding, cost function evaluation, tournament selection, uniform crossover, elitism and convergence checking with re-initialization. A through review of implementing Micro GA is given in Senecal [18] and Kazarlis et al. [19]. However the basic functional components are explained for the sake of completeness.

4.2.1. Variable encoding and decoding

For convenient implementation of the basic genetic operators of the GA, the parameters are coded into a binary string. (Goldberg [14]). The binary string is commonly referred to as a *chromosome*, while its features are known as *genes*. The precision of each of the design parameters included in a chromosome is based on the desired range of the parameter (i.e., its minimum and maximum values) and how many bits are used in its representation. Thus, the precision π of an input parameter X_i is given by (Homaifar et al. [20])

$$\pi = \frac{X_{max} - X_{min}}{2^{\lambda} - 1} \tag{12}$$

where λ is the number of bits used to encode the parameter.

For example, the y coordinate of the mid point of the heat source has the extreme values as $y_{min} = 0.02$ m and $y_{max} = 0.38$ m. As 16 bits are used to encode this variable to the binary form, the precision of y is 5.49×10^{-6} m. This high precision is required for adequate sampling of the search space, thus avoiding under sampling.

4.2.2. Cost function evaluation

As discussed earlier, the cost function (C) for a particular set of input variables is determined from a numerical simulation using the CFD solver. With an appropriate grid, simulations were performed in the CFD solver to evaluate the maximum temperature (T_{max}) on any of the three discrete heat sources. T_{max} is then used to evaluate the cost function (C) from Eq. (11).

4.2.3. Tournament selection

Survival of the fittest translates into discarding the chromosomes with the highest cost and selecting the chromosomes with the lowest cost. Tournament selection is one

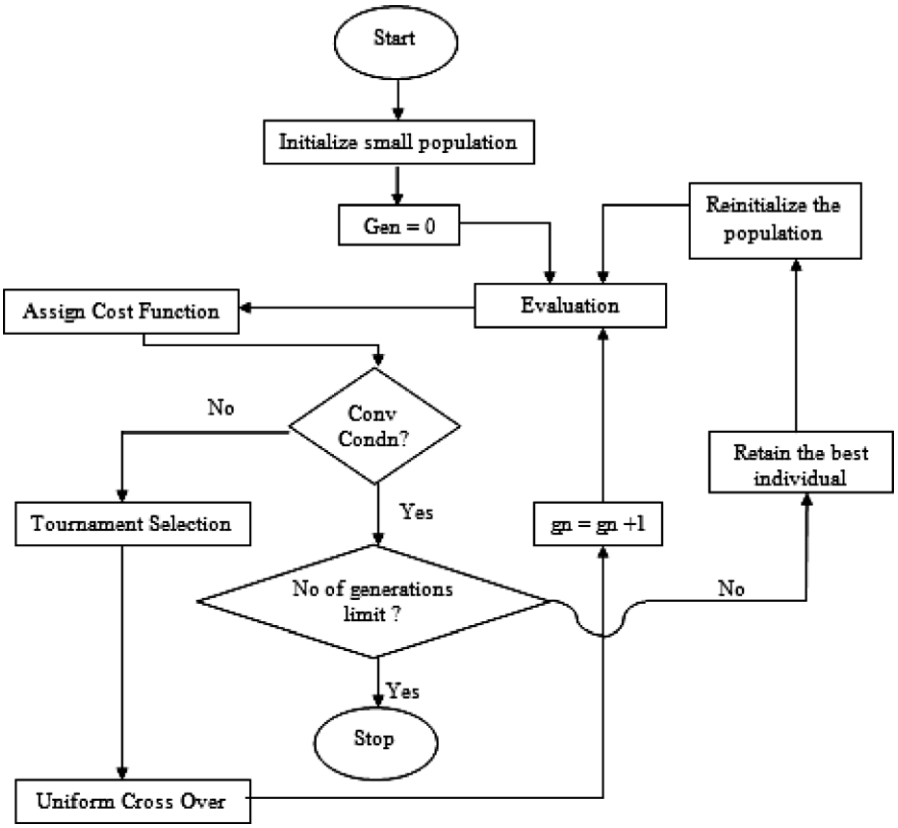

Fig. 5. Flow chart of functional components of Micro GA used in the present study.

of the popular methods for selection in micro genetic algorithm as it ensures better diversity in the small micro populations of five individuals. Initially ($N_{pop}/2$) groups of individuals are randomly selected from the total N_{pop} population without replacement. These groups participate in a tournament where the winning individuals of each tournament, with lower cost value, will form the $N_{pop}/2$ mating pool. This process is repeated twice to create the N_{pop} mating pool.

4.2.4. Uniform crossover

Uniform crossover is one of the methods of reproduction wherein bits are exchanged between two string individuals. In particular, two individuals are randomly selected as parent individuals from the mating pool based on tournament selection. Then arbitrary positions on both individuals are chosen for crossing locations where exchange of bits takes place. A crossing mask is employed to determine the crossing locations. Two parent individuals will exchange their bits where the corresponding value in mask is one. The mask consists of 0 s and 1 s distributed randomly across its length. In this study, the crossover probability, which is percent of bits exchanged, is set at 100%.

4.2.5. Elitism

In addition to performing the cost function evaluation, tournament selection, uniform crossover, MGA uses the elitism. Elitism guarantees that the best string individual survives until the last generation. More specifically, if the best offspring individual is worse than the best parent individual, the best parent individual will randomly replace any offspring individual. That is, the offspring individuals in the current generation become parent individuals in the next generation.

4.2.6. Convergence

Convergence of a genetic algorithm is evaluated based on two factors.

1. *Population convergence*: Population convergence is defined as the progression towards chromosome uniformity. Thus, for the genetic algorithm operators described above, a gene may be considered to be converged, when 95% of that particular gene in the entire population shares the same value [19]. A population is then converged when all of the genes are converged. When a population has converged, the best individual is stored and the rest are re-initialized to bail the solutions out from the local minima towards the global optimum, as there is no mutation in the micro genetic algorithm.
2. *Algorithm convergence*: There are several different methods to specify the convergence criteria for the genetic algorithm like specifying a threshold for the minimum cost function, specifying a threshold for the difference between the lowest and the mean cost in the population, specifying a maximum number of generations to be evolved. In the current study, maximum number of generations has been set to 1000 and the difference between the best and average fit individuals is set as 5% of the best cost function value.

4.3. Validation of the computational code for MGA

Based on all the above factors and according to the flow chart given in Fig. 5, a computational code for optimizing using the Micro GA has been developed using the commercial Matlab 7.0 software. To ensure the validity of the code developed, it was tested with Rastrigin's function, Ras(x). This function is widely used as a benchmark for testing optimization algorithms as it has many local minima that make it difficult for standard, gradient-based methods to find the global minimum

$$\mathrm{Ras}(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2) \tag{13}$$

Despite its complexity, the function has just one global minimum, which occurs at the point [0, 0] in the x_1–x_2 plane, where the value of the function is 0. At any local minimum other than [0, 0], the value of Rastrigin's function is greater than 0. The farther the local minimum is from the origin, the larger the value of the function is at that point.

Fig. 6 shows the optimization of Rastrigin function with the developed micro genetic algorithm code. It can be observed that the optimum is reached very fast within 80 generations and the optimum value is found to be exactly equal to the global optimum value of zero. This clearly validates the MGA code for solving complex optimization problems. Hence the code has been used in the present study to find the optimum location of the heat sources.

4.4. Solution procedure

The solution procedure for optimizing the heat source location is illustrated in the flow chart shown in Fig. 7.

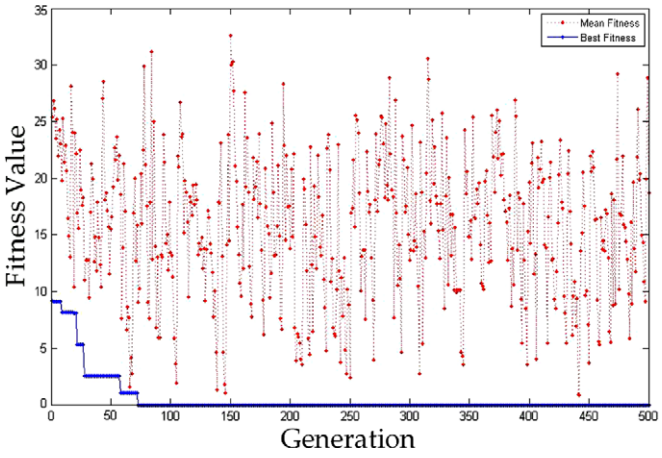

Fig. 6. Determining the global minimum of Rastrigin function using the Micro GA code.

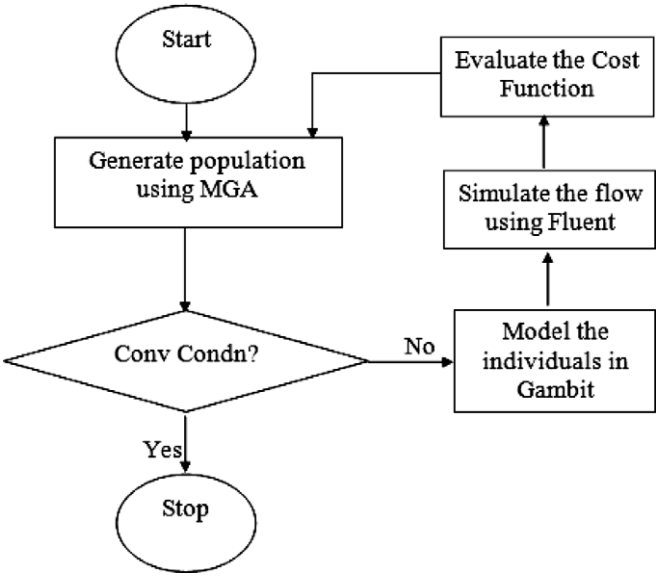

Fig. 7. Flow chart illustrating the optimization of location of heat sources.

The various steps involved are as follows:

(1) Initially a population of five individuals is randomly generated and it is then checked for population convergence.
(2) If the population of individuals has not converged, then all the five cases are modeled using the meshing software to create a sufficiently refined mesh to ensure a low discretization error.
(3) These five cases with different configurations of heat sources are then simulated in the solver and the maximum temperature (T_{max}) on any of them is evaluated.
(4) T_{max} is used to calculate the cost function values using Eq. (11).
(5) These cost function values are passed to the micro genetic algorithm which applies the genetic operators and creates the next generation.

These sequences of steps are repeated until the micro genetic algorithm converges to the global optimum. In this manner, the optimization has been done for 66 generations and the results are shown in Fig. 8. It can be observed that for the first 20 generations, lowest cost (C) = -740 which corresponds to a temperature of 355.97 K. Micro GA finds another local optimum after 20 generations with a cost function value $C = -979.6$, temperature of 342.18 K. The average cost value is fluctuating because MGA is constantly searching in different regions in the search space to find better solutions.

4.5. Limitations

Time was the major constraint for the simulation. The grid for the model had 25,000 nodes and the simulations were done using Fluent 6.3 software on an IBM p690 series machine, a 32 Processor machine with a RAM size of

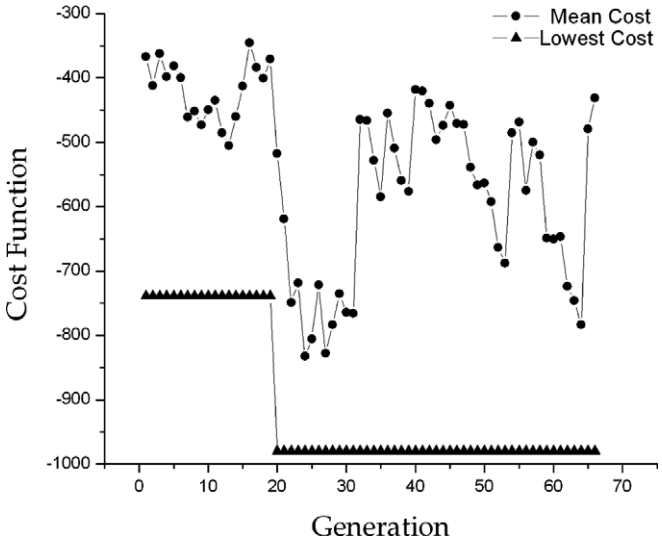

Fig. 8. Variation of cost function with generation.

64 GB. Under such conditions, simulation for each individual in a generation took around 6 hours which shows that the simulation is a computationally intensive process when the solutions have to be obtained repeatedly. In order to reduce the time taken, Fluent parallel solver was employed wherein the grid was split into four partitions and distributed to 2 computer nodes for parallel processing. The time taken for each simulation was reduced by half to 3 h, which was still considerably high. Hence, in order to overcome this severe constraint on time and to obtain the global optimum, a forward model should be developed to predict the cost function values based on the six coordinates of the heat source locations.

5. Artificial neural networks

Artificial neural networks (ANNs) are computational modeling tools that have recently emerged and found extensive acceptance in many disciplines for modeling complex real-world problems. ANNs may be defined as structures comprised of densely inter-connected adaptive simple processing elements (called artificial neurons or nodes) that are capable of performing massively parallel computations for data processing and knowledge representation. The attractiveness of ANNs comes from the remarkable information processing characteristics such as non-linearity, high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information and their capability to generalize. ANN-based models are empirical in nature; however they can provide practically accurate solutions for precisely or imprecisely formulated problems and for phenomena that are only understood through experimental data and field observations.

5.1. Artificial neuron model

Artificial neural networks are inspired by models of living neurons. Artificial neurons are nodes in a neural net-

work, and these nodes are the processing units that perform nonlinear summing functions. Fig. 9 shows a typical artificial neuron which receives input signals s_i from possibly n different sources. These signals traverse weighted pathways w_{ij}, in order to generate the internal activation x_j, which is a linear weighted aggregation of the impinging signals, modified by an internal threshold θ_j. The activation x_j is given by Eq. (14)

$$x_j = \sum_{i=1}^{n} w_{ij} s_i + \theta_j \tag{14}$$

The activation of a neuron is subsequently transformed through a signal function δ, to generate the output signal $s_j = \delta(x_j)$. The sigmoid signal function involves an exponential function to generate the output from the neuron activation. The equation for this function is given as follows:

$$\delta(x_j) = \frac{1}{1 + e^{-x_j}} \tag{15}$$

This function, shown in Fig. 10, has properties like monotony and continuity which enable the networks to approximate and generalize on functions by learning from data. Hence, this signal function is most widely used in the hidden layers of the neural network.

5.2. Back-propagation ANN

Artificial neurons are grouped together to form a layer of neurons. An artificial neural network (ANN) consists of several layers of neurons to train the network for function approximation or generalization. The feed forward

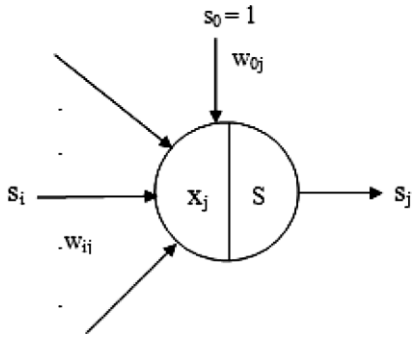

Fig. 9. Illustration of an artificial neuron.

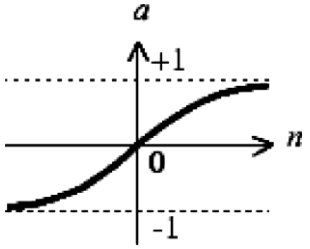

Fig. 10. Sigmoid signal function of an artificial neuron.

Back-Propagation algorithm is the most widely used method to train the ANNs. Back-propagation (BP) is based on searching an error surface (error as a function of ANN weights) using the gradient descent algorithm for points with minimum error. Each iteration in BP constitutes two steps.

5.2.1. Forward sweep

In an initialized ANN (i.e., an ANN with assumed initial weights), the forward sweep involves presenting the network with one training example. This starts at the input layer where each input node transmits the value received forward to each hidden node in the hidden layer. The collective effort on each of the hidden nodes is summed up using Eq. (14) to evaluate the activation at each node. Once the activation at the hidden node is determined, the output signal at that node is evaluated using the neuron signal function. The same procedure of calculating the net effect is repeated for each hidden node and for all hidden layers. The net effects calculated at the output nodes are consequently transformed into output signal using a neuron signal function. This output signal is the ANN solution of the fed example, which may deviate considerably from the target solution due to the arbitrary selected interconnection weights.

5.2.2. Backward sweep

In the backward sweep, the difference (i.e., error) between the ANN and the target outputs is used to adjust the interconnection weights, starting from the output layer, through all the hidden layers, to the input layers. The weights are adjusted using the gradient descent algorithms like the Quasi Newton, Resilient back-propagation, Levenberg–Marquardt and so on.

5.3. Generalization of ANN

Generalization is the property of artificial neural networks to give reasonable answers on unseen input parameter combinations. If a network is trained with only few training samples, then it may quickly 'over fit' the training data which means that the network error is driven to a small value for the training samples but will become large when a new input is presented. This indicates that the network has memorized the training samples which seriously affect the generalization property of the ANN. There are two popular methods to avoid over fitting and improving the generalization performance.

5.3.1. Early stopping

In this technique, the available data is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error.

However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations the training is stopped, and the weights and biases at the minimum of the validation are returned. The biggest disadvantage of early stopping is that it requires the data set to be divided into training, validation and test thereby not using all the available data.

5.3.2. Bayesian regularization

Bayesian regularization works by modifying the performance function by adding a term that consists of the mean of the sum of squares of the network weights. This modifies the performance function such that it incorporates a bias against large network weights. The performance function $F(w)$ is given by Eq. (16)

$$F(w) = \alpha E_{\mathrm{D}}(w) + \beta E_{\mathrm{w}}(w) \tag{16}$$

The sum squared error $E_{\mathrm{D}}(w)$ and weight decay regularizer $E_{\mathrm{w}}(w)$ are defined by Eqs. (17) and (18)

$$E_{\mathrm{D}}(w) = \frac{1}{2}\sum_{i}^{n}(a_i - t_i)^2 \tag{17}$$

$$E_{\mathrm{w}}(w) = \frac{1}{2}\sum_{i,j}^{n} w_{ij}^2 \tag{18}$$

where n is the total number of training patterns, a_i is the output of the ANN and t_i is the target available data.

David MacKay [21] had first proposed a probabilistic approach to the ANN learning process. In the Bayesian framework, the network weights are assumed to be random variables. The objective of the Bayesian training process is to find the most probable weights (w) and the regularization parameters α and β such that the performance function $F(w)$ given by Eq. (16) is minimized.

According to Bayes rule the probability distribution can be written as

$$P(w|D,\alpha,\beta,\mathrm{Mod}) = \frac{P(D|w,\alpha,\mathrm{Mod})P(w|\beta,\mathrm{Mod})}{P(D|\alpha,\beta,\mathrm{Mod})} \tag{19}$$

where D corresponds to the input–output data samples, Mod denotes the network model and architecture. $P(w|\beta,\mathrm{Mod})$ is the prior distribution, which corresponds to our knowledge of the weights before any data is collected. $P(D|w,\alpha,\mathrm{Mod})$ is the likelihood of the data occurring, given the weights w. The error function is now interpreted as defining the probability distribution of a noise model

$$P(D|w,\alpha,\mathrm{Mod}) = \frac{1}{Z_{\mathrm{D}}(\alpha)}\mathrm{e}^{-\alpha E_{\mathrm{D}}} \tag{20}$$

where $Z_{\mathrm{D}}(\alpha) = (\Pi/\alpha)^{N/2}$ and N is the number of network outputs. So the use of the sum squared error E_{D} corresponds to an assumption of Gaussian noise on the target variables. Similarly, the regularizer $E_{\mathrm{W}}(w)$ is interpreted in terms of a prior distribution over the parameters

$$P(w|\beta,\mathrm{Mod}) = \frac{1}{Z_{\mathrm{D}}(\beta)}\mathrm{e}^{-\beta E_{\mathrm{w}}} \tag{21}$$

Substituting the expressions for the likelihood function and prior probability into Eq. (19) gives the posterior distribution of the weights shown in Eq. (22)

$$P(w|D,\alpha,\beta,\mathrm{Mod}) = \frac{1}{Z_{\mathrm{F}}(w)}\mathrm{e}^{-F(w)} \tag{22}$$

Applying Bayes theorem, the probability distribution of α, β is given by Eq. (23)

$$P(\alpha,\beta|D,\mathrm{Mod}) = \frac{P(D|\alpha,\beta,\mathrm{Mod})P(\alpha,\beta|\mathrm{Mod})}{P(D|\mathrm{Mod})} \tag{23}$$

The optimum regularization parameters are calculated by maximizing the posterior probability $P(\alpha,\beta|D,\mathrm{Mod})$.

Bayesian regularization has the specific advantage that it does not require any 'test set', 'validation set', hence all the available data can be used for training. The salient advantage of this algorithm is that it implicitly provides a measure of how many network parameters (weights) are effectively used by the network. This indicates whether a sufficient number of training samples are presented to the network and automatically determines the optimum network size.

5.4. Network architecture

There are four elements that comprise the artificial neural network's architecture:

1. The training algorithm.
2. The activation functions at each layer.
3. The number of layers.
4. The number of neurons in each layer.

Based on the discussion in Section 5.3, Bayesian regularization is employed as the training algorithm to train the network. Due to the inherent advantages of continuity and monotonic nature, tangential sigmoid function is used as the activation function for all the hidden layer neurons and linear function is used in the output layer. Since there are six inputs to the neural network, it has six neurons in the input layer and one neuron in the output layer. There are no established methods to objectively find the number of hidden layers and the neurons in the hidden layers; hence they have been found out by trial and error.

The 66 generations of Micro GA optimization provide 238 distinct input data sets. Eighty percent of the data is randomly selected and used for training the network and remaining 20% is used to test the network performance. The following are the performance parameters defined in Ermis et al. [22] calculated on the test data and used to compare the different network architectures.

1. Mean relative error (MRE)

$$\mathrm{MRE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i - t_i|}{|t_i|} \tag{24}$$

2. Absolute fraction of variance (R^2)

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{n}(a_i - t_i)^2}{\sum_{i=1}^{n}(t_i)^2}\right] \quad (25)$$

3. Correlation coefficient (R_{test}) between the ANN predicted values and the actual values in the test data set.

The performance parameters on various networks have been summarized in Table 3. It can be observed that for a network with single hidden layer, minimum error occurs when there are 7 hidden neurons and the error on the test data increases with the addition of more neurons. The data clearly shows that for a network 11, MRE has a least value of 14.3% and R_{test} is 0.93. Hence, based on the above discussion, a neural network with seven neurons in the first hidden layer and 2 neurons in second hidden layer is used in the present study. Fig. 11 shows the neural network architecture with the six input variables and the Cost function (C) as the output.

Table 3
Performance parameters on different networks

S. No.	Number of neurons in layer 1	Number of neurons in layer 2	MRE	R^2	R_{test}
1	6	0	17.192	0.971	0.904
2	7	0	14.698	0.976	0.934
3	8	0	16.054	0.975	0.912
4	10	0	19.024	0.956	0.866
5	12	0	16.065	0.968	0.949
6	15	0	16.859	0.976	0.910
7	20	0	16.935	0.967	0.882
8	30	0	16.558	0.964	0.854
9	6	2	14.316	0.960	0.892
10	6	6	15.970	0.971	0.900
11	7	2	14.298	0.978	0.928
12	7	5	17.432	0.958	0.873

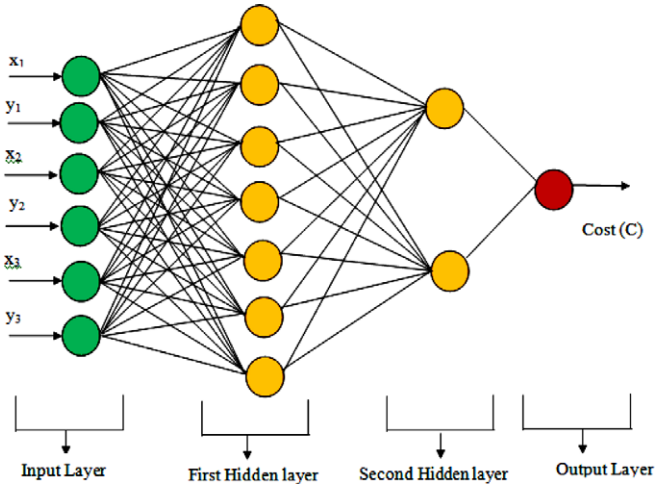

Fig. 11. Neural network architecture employed in the present study to predict the cost function values from six inputs.

6. Results and discussion

6.1. Integration of the ANN with Micro GA

As discussed in the earlier chapters, in order to overcome the severe time constraint, a forward model had to be developed. For this purpose, the artificial neural network (ANN) shown in Fig. 11 is trained to predict the cost function values from the six inputs corresponding to the heat source locations within a reasonable range of accuracy. This ANN is the forward model and supplants the numerical simulations, thereby reducing the time required for evaluating one case from 3 h to 1 s. This phenomenal reduction in time taken facilitates the Micro Genetic Algorithm to optimize faster.

After integration of the ANN with Micro GA, the algorithm was allowed to run until convergence to evaluate the global optimum and the results are shown in Fig. 12. It shows that after successive intervals, the lowest cost continues to decrease. The Micro GA algorithm employed evolved the population of individuals for 1000 generations as shown in Fig. 12. After 750 generations, the MGA has located the optimum with a cost function of −1508.36 which corresponds to a temperature of 327.39 K. As a test of this optimum, this best individual was again numerically simulated using the solver and T_{max} was determined to be 329.65 K which corresponds to an error of only 0.7% compared to the ANN's predicted temperature. In addition, even when the population was made to evolve for 5000 generations, the above optimal solution did not change further which shows that it is indeed the "true global optimum".

The location of the heat sources for the best individuals are given in Table 4. All the temperatures are very close to each other varying from 329 K to 335 K which shows that micro GA gives many optimal/near optimal solutions, and is "egalitarian" so to speak, while the traditional gradient descent algorithms provide only one optimum.

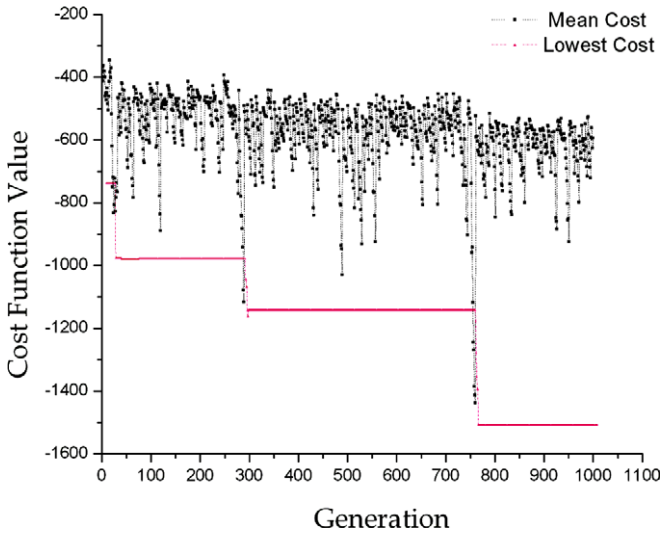

Fig. 12. Optimization of heat sources using micro GA.

Table 4
Optimal locations of the best individuals

T_{max} (K)	x_1 (cm)	y_1 (cm)	x_2 (cm)	y_2 (cm)	x_3 (cm)	y_3 (cm)
329.65	35.88	19.74	37.94	31.65	38.20	24.24
330.22	36.91	31.65	34.60	20.00	36.14	19.47
332.17	36.91	31.65	34.60	20.00	38.20	22.38
335.17	33.82	31.65	35.88	20.00	38.20	22.38
335.63	33.82	31.65	35.88	20.00	38.20	22.38

6.2. Profile contours of the optimal distribution of the heat sources

6.2.1. Velocity

The fluid flows almost horizontally at the inlet due to the high velocity of 4 m/s. As it approaches the opposite wall, it is tilted upwards and moves vertically towards the outlet. On reaching the outlet, some amount of fluid turns in the opposite direction and moves horizontally near the top wall and vertically near the left wall, thereby creating a vortex as seen in Fig. 13. It can also be observed that the fluid at the bottom right and top left corner is almost static. So to avoid this, a small tilt at the bottom right (instead of a sharp corner) may be employed to guide the fluid in the right direction. Figs. 14 and 15, respectively show horizontal velocity and vertical velocity profiles which clearly indicate the vortex formations.

6.2.2. Temperature

The temperature contours for the optimal configuration are shown in Fig. 16. It shows that the temperature in most of the cavity region is not affected by the heat sources. The most interesting feature of this optimal solution is that, all the heat sources are located along the flow streamlines i.e., no heat source blocks the flow; therefore there is no wake region created. During the MGA iterative process, it has been observed that if a heat source falls in the wake region of another heat source, then its temperature would be as

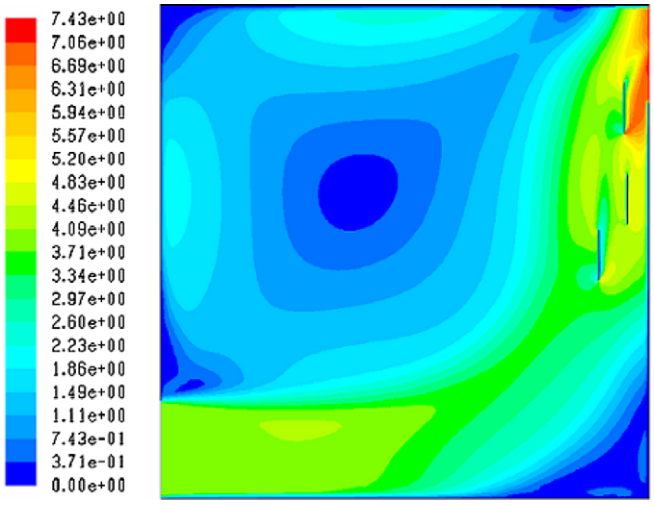

Fig. 13. Velocity magnitude profile of the optimal distribution of heat sources in the cavity (m/s).

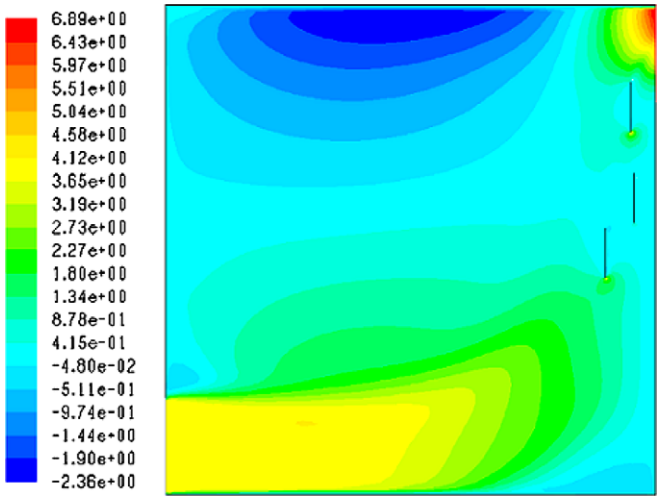

Fig. 14. X-velocity (u) Profile of the optimal distribution of heat sources inside the cavity (m/s).

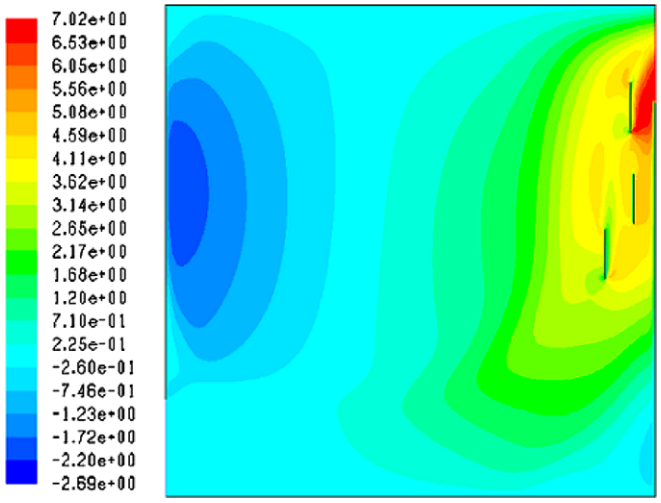

Fig. 15. Y-velocity (v) profile of the optimal distribution of heat sources inside the cavity (m/s).

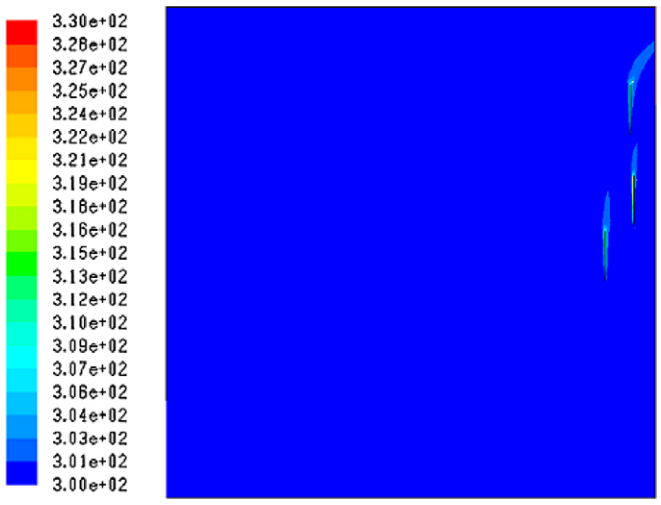

Fig. 16. Temperature profile of the optimal distribution of heat sources inside the cavity (K).

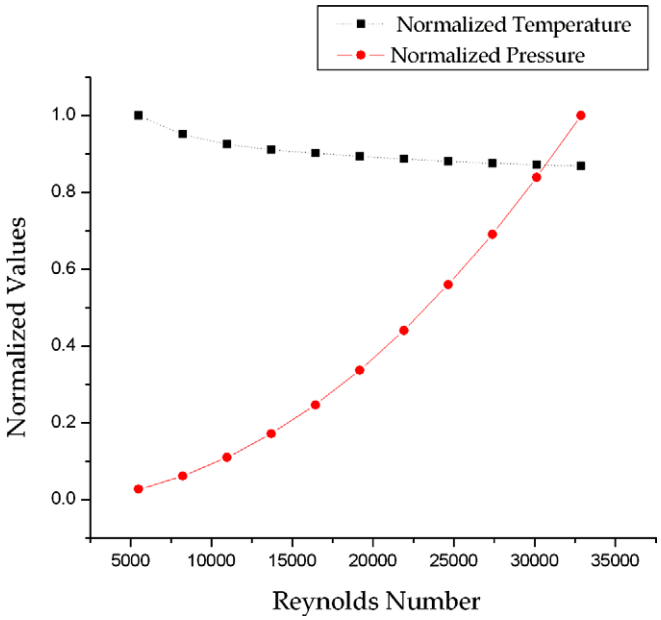

Fig. 17. Sensitivity analysis of the optimal distribution of heat sources.

high as 500 K. Such high temperatures are explained based on knowledge that in the wake region velocities are lower and the fluid temperature is higher leading to a lower heat transfer coefficient.

6.3. Sensitivity studies

Sensitivity studies have been done on the optimal distribution of heat sources by varying the Reynolds number (Re) from 5000 to 35,000. This is done to evaluate the change in parameters like maximum temperature (T_{max}) and the maximum pressure drop (Δp_{max}) with an increase in velocity. The normalized values of temperature and pressure drop, defined by Eqs. (26) and (27), are plotted against Reynolds number in Fig. 17.

$$\text{Normalized temperature } T^*_{max} = \left[\frac{T_{max}}{T^{max}_{max}}\right] \tag{26}$$

$$\text{Normalized pressure drop } \Delta p^*_{max} = \left[\frac{\Delta p_{max}}{\Delta p^{max}_{max}}\right] \tag{27}$$

where T^{max}_{max} and Δp^{max}_{max} are the maximum temperature and pressure drop among the data obtained by varying the Reynolds number.

Fig. 17 shows that as the Reynolds number is increased, there is a variation of 10% in the normalized temperature for the Reynolds number ranging from 5000 to 15,000, while there is no appreciable change in temperature at Reynolds number higher than 15,000. On the other hand, the maximum pressure drop increases very rapidly. Therefore, the pressure drop is a more severe constraint in the design of thermal systems than the heat transfer as the power of the intake fan has to be varied accordingly to overcome the pressure drop.

7. Conclusions

In this study, Bayesian regularization neural network, trained with sufficient number of data samples evaluated from a finite volume solver, is used as a forward model for cost function evaluation in the Micro genetic algorithm optimization of the location of multiple discrete heat sources in a ventilated cavity. The salient advantage of this method is that, it requires less function evaluations from the finite volume solver than a conventional optimization algorithm needs, hence the computational time can be reduced substantially which will help in obtaining optimal solutions faster.

The optimal distribution of three discrete heat sources is when all of them are along the flow streamlines i.e., no heat source is blocking the flow to create a wake region behind it. From the sensitivity analysis, it has been observed that as the Reynolds number is increased for the optimal distribution, the maximum temperature T_{max} falls slowly while the maximum pressure drop increases rapidly indicating that pressure drop is a stronger constraint in the design of thermal systems rather than heat transfer.

The present study has only dealt with reducing the temperature in the optimization process. However, in practical situations both pressure drop and temperature should be considered in the design process. This can be done by using multi-objective genetic algorithms.

References

[1] Tunc Icoz, Nitin Verma, Yogesh Jaluria, Design of Air and Liquid Cooling Systems for Electronic Components Using Concurrent Simulation and Experiment, J. Elect. Packag. 128 (2006) 466–479.
[2] H. Bhowmik, C.P. Tso, K.W. Tou, Analysis of convection heat transfer from discrete heat sources in a vertical rectangular channel, Trans. ASME J. Heat Transfer 127 (2005) 215–222.
[3] Roy. N. Mathews, C. Balaji, Numerical Simulation of conjugate turbulent mixed convection heat transfer in a vertical channel with discrete heat sources, Int. Commun. Heat Mass Transfer 33 (2006) 908–916.
[4] C.Y. Choi, A. Ortega, Mixed convection in an inclined channel with discrete heat sources, 1992 Inter Society Conference in Thermal Phenomenon, pp. 40–48.
[5] A.K. da Silva, S. Lorente, A. Bejan, Optimal distribution of discrete heat sources on a plate with laminar forced convection, Int. J. Heat Mass Transfer 47 (2004) 2139–2148.
[6] A.K. da Silva, S. Lorente, A. Bejan, Optimal distribution of discrete heat sources on a wall with natural convection, Int. J. Heat Mass Transfer 47 (2004) 203–214.
[7] Tito Dias Jr., Luiz Fernando Milanez, Optimal location of heat sources on a vertical wall with natural convection and genetic algorithm, Int. J. Heat Mass Transfer 49 (2006) 2090–2096.
[8] A. Bejan, S. Lorente, Constructal theory of generation of configuration in nature and engineering, J. Appl. Phys. 100 (2006) 041301.
[9] Fluent Inc, Computational fluid dynamics software, Lebanon, NH.
[10] Christopher J. Roy, Review of code and solution verification procedures for computational simulation, J. Comput. Phys. 205 (2005) 131–156.
[11] F. Ampofo, T.G. Karayiannis, Experimental benchmark data for turbulent natural convection inside a air filled square cavity, Int. J. Heat Mass Transfer 46 (2004) 3551–3572.

[12] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.
[13] J.H. Holland, Adaption in Natural and Artificial systems, University of Michigan Press, Ann Arbor, 1975.
[14] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Massachusetts, 1989.
[15] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Veechi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[16] M. Dorigo, G. Maria, Ant colony system: a cooperative learning approach to the travelling salesman problem, IEEE Trans. Evolut. Comput. 1 (1997) 53–66.
[17] K. Krishnakumar, Micro genetic algorithms for stationary and non-stationary function optimization, SPIE 1196, Intelligent Control and Adaptive Systems, 1989.
[18] Peter Kelley Senecal, Numerical Optimization using the gen4 micro genetic algorithm, Engine Research Centre, University of Wisconsin-Madison, 2000.
[19] S.A. Kazarlis, S.E. Papadakis, J.B. Theocharis, V. Petridis, Micro genetic algorithms as generalized hill – climbing operators for optimization, IEEE Trans. Evolut. Comput. 5 (2001) 204–217.
[20] A. Homaifar, H.Y. Lai, E. McCormick, System optimization of Turbo fan engines using genetic algorithm, Appl. Math Model. 18 (1994).
[21] D.J.C. MacKay, Bayesian method for adaptive models, Ph.D. Thesis, California Institute of Technology, 1991.
[22] Kemal Ermis, Aytunc Erek, Ibrahim Dincer, A report on Heat Transfer analysis of phase change processes a finned-tube thermal energy storage system using artificial neural network, Int. J. Heat Mass Transfer 50 (2007) 3163–3175.