Optimal Modeling of Urban Ambient Air Ozone Concentration Based on Its Precursors' Concentrations and Temperature, Employing Genetic Programming and Genetic Algorithm

SEYED MAHMOUD MOUSAVI[+], DANIAL HUSSEINZADEH AND SADEGH ALIKHANI

Efficient models are required to predict the optimum values of ozone concentration in different levels of its precursors' concentrations and temperatures. A novel model based on the application of a genetic programming (GP) optimization is presented in this article. Ozone precursors' concentrations and run time average temperature have been chosen as model's parameters. Generalization performances of two different homemade models based on genetic programming and genetic algorithm (GA), which can be used for calculating theoretical ozone concentration, are compared with conventional semi-empirical model performance. Experimental data of Mashhad city ambient air have been employed to investigate the prediction ability of properly trained GP, GA, and conventional semi-empirical models. It is clearly demonstrated that the in-house algorithm which is used for the model based on GP, provides better generalization performance over the model optimized with GA and the conventional semi-empirical ones. The proposed model is found accurate enough and can be used for urban air ozone concentration prediction.

Key words: *Genetic programming, genetic algorithm, semi-empirical models, optimization, ozone concentration, urban ambient air*

1. Introduction

The main objective of system modeling[1] is the development of rules expressing the relation between quantities known as priori and other quantities that are directly correlated with system behavior. These rules are perceived as a model of the system. Various model classification criteria can be distinguished while dealing with a system modeling task. These different types of models translate into different modeling requirements.

Completeness and harmony in nature are largely the result of evolutionary forces that have adapted species to their surroundings and to each other. Examining natural phenomena, one can appreciate the potential of nature's ready-made solutions, which are often more efficient than man-made ones. For example, consider spider silk, which is more elastic than nylon and stronger than steel, and witness the abilities of the spider, essentially blind with a limited nervous system, to use six variants of silk to build a robust, complex structure in an unpredictable environment[2]. With nature as a motivator, recent decades have seen increasing attempts to mimic natural evolution using computers[3-5]. These efforts are stimulated by Darwin's notion of "the survival of the fittest", are generally referred to as evolutionary-computation, and have been used successfully to solve particularly complex problems.

The most commonly used evolutionary-computation algorithm is the genetic algorithm, which is based on solutions of fixed-length chromosomes, usually consisting of binary genes, organized into sequences, often termed schema[4].

Genetic algorithms, mostly referred to as GAs, are one of the optimization methods finding wide application in optimization problems. Genetic algorithm searches stochastically through the real space of problem by generating some of the potential solutions, which is named as initial population. This method is a kind of evolution-based systems that measures the fitness of each individual in population, and then selects the fittest individuals until reproducing the intermediate population. Recombinant genetic operators affect some individuals in this population and produce the next population for the new generation. The GA could optimize linear and nonlinear objective functions by exploring the space of problem and exponentially exploiting promising areas through selection, crossover, and mutation operations applied to individuals in the population[6]. Based on the advantages and limitations of genetic algorithm, it could be realized when

Department of Chemical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[+]Corresponding author : e-mail : mmousavi@um.ac.ir

genetic algorithm should be applied for optimization problems. Parallelism (solve the problem by several chromosomes), wide solution space, complexity of fitness function, discovering the global optimum, using as multi-objective function, without gradient complexity and just using function evaluations the GA is easily modified for different problems. Moreover, it handles noisy functions well and is resistant to becoming trapped in local optima. But it should be considered that the GA has some limitations like identifying fitness function, definition of representation, premature convergency, problem of choosing the various parameters like the size of the population, mutation rate, crossover rate, the selection method and number of elites and sometimes needs to be coupled with a local search technique[7, 8].

In spite of being efficient in a number of fields, genetic algorithms are unsuitable for creating empirical model structures, since they manipulate populations of solutions of fixed-length chromosomes, while the optimal complexity of empirical models is unknown in advance. Because of this perceived need for more intelligent construction of empirical models, a new family of evolutionary computation methods has emerged, based on established GA ideas. These new algorithms, referred to as genetic programming, rely on tree-like building blocks, and therefore support populations of model structures of varying length and complexity[9]. Activity in genetic programming was introduced by Koza[5], who demonstrated its applications in fields such as robotics, games, control, and symbolic regression.

One of the important applications of genetic programming is in creating input–output empirical models in process system engineering (PSE) applications[10]. The class of empirical models can be divided into two broad categories: (a) models with predefined structure (either linear or nonlinear), and whose parameters are determined to maximize the capacity of process data prediction; or (b) black-box models with undetermined structure. An example of the first category would be a linear model relating a dependent variable, y, to a set of n independent variables, u_i[9]:

$$y = \sum_{i=1}^{n} a_i u_i \tag{1}$$

where the coefficients a_i are determined to maximize the predictive power of the model.

An example of a black-box model would be an artificial neural network (ANN), in which the number and identity of the relevant inputs and the number of layers are the only attributes of the structure that are determined by the user. In present research, the first category of aforementioned models was employed to apply GP.

The functional relationships, observed or theoretically derived, between urban ozone and its precursors, non-methane hydrocarbons (NMHC) and nitrogen oxides ($NO_x = NO + NO_2$), have been reviewed in criteria document[11]. Air-quality simulation models (AQSM) represent the most fundamental approach to relate precursor emissions to ozone air quality. However, there are still significant uncertainties in results obtained from AQSM because of large uncertainties in the AQSM input data as well as the difficulties in representing atmospheric processes accurately. The derivation of semi-empirical models based on ambient and smog chamber data is another approach to the ozone-precursor relationship. These semi-empirical models could supplement results based on AQSM and could provide useful insight into the ozone-precursor relationship relevant to ozone control strategies.

Several attempts have been done to derive semi-empirical models based on ambient data[12]. However, the variability of ozone is generally dominated by meteorological variables, particularly temperature. As a result, it is difficult to discern the relationship of ozone to NMHC and NO_x. Moreover, available ambient NMHC data are very limited.

Sillman investigated the relation between ozone, NO_x, hydrocarbons and other factors in urban and polluted rural environments[13].

During the 1987 Southern California Air Quality Study (SCAQS)[14], extensive outdoor smog chamber experiments were performed on Los Angeles air by the General Motors Research Laboratories in order to test and clarify some models.

In this paper, a general, but simple ozone-precursor relationship is investigated using experimental data. In-house software was also prepared in this research for both GP and GA, based on learning algorithms presented in the next section. The model predictions were verified using ozone and its precursors' concentration experimental data. The objective of present work is to investigate the validation of current semi-empirical models, compare obtained results from semi-empirical models with those of purely empirical models expressed and finally, propose a reliable model to predict ozone concentration based on its precursors' concentration.

2. Materials and methods

2.1 Semi-empirical models

Based on smog chamber studies of the olefin-NO_x system, Akimoto et al.[15] proposed that the following relationship was held in the NMHC-excess region:

$$(O_3) = \alpha (O_3)_{ps} \qquad (2)$$

where (O_3) is the maximum ozone concentration reached, α is the proportionality constant and $(O_3)_{ps}$ is the photostationary ozone concentration in the absence of NMHC. Shen and Springer[16], on the basis of flow reactor studies of the cyclohexene-NO_2 system, proposed the following relationship:

$$(O_3) = [k_1 (NO_x)]^{1/2} F(R) \qquad (3)$$

where k_1 is the photolysis rate constant of NO_2, (NO_x) is the initial concentration of NO_x, R is the initial NMHC/NO_x ratio and F(R) represents a function of R only. Since $(O_3)_{ps}$ is proportional to $[k_1 (NO_x)]^{1/2}$ when (NO_x) is not too low (≥ 0.01 ppm)[15], equation 2 is a special case of Eq. (3) where F(R) becomes constant.

Kelly and Gunst[17] showed that, in irradiations of identical NMHC/NO_x mixtures, ozone maximum concentrations are strongly dependent on the average temperature, and that, from multiple linear regression modeling, temperature and initial NMHC and NO_x concentrations are adequate to represent ozone maximum concentrations. They derived the following empirical model:

$$(O_3) = 0.129 + (NO_x) (2.8 - 6.8/R) + 0.019(T_{av} - 21.1) \qquad (4)$$

where concentrations are in ppm, T_{av} is the run-average temperature in °C, and 21.1 °C is the average of the daily run-average temperatures.

Chang and Rudy[18], based on a simulation of smog chamber experiments with initial input of precursors, but with no emissions (no additional input of precursors) and no dilution searched functional forms similar to Eq. (3), emphasizing simplicity in the functional form. After a number of trials, the following functional form was chosen:

$$(O_3) = c + \gamma (NO_x)^{1/2}[1 - exp(-aR^b)] \qquad (5)$$

Parameters c, γ, a and b are determined by a non-linear least-square method using a non-linear function minimization routine[19]. Parameter c is used to account for aloft polluted air being entrained. In order to include the temperature effect, Eq. (5) was changed to the following functional form[18]:

$$(O_3) = c + \gamma (NO_x)^{1/2}[1 - exp(-aR^b)]exp[-d(1/T - 1/T_{av})] \qquad (6)$$

Parameters c, γ, a, b and d are determined by a non-linear regression described earlier. Parameter c was added to represent the smog chamber wall effects in the studied case. T is the run-average temperature for each run, and T_{av} is the average of daily run-average temperatures.

The obtained results from equations (4) and (6) represented the experimental data reasonably. Because of having more adjustable parameter, Eq. (6) is more accurate. The mentioned model (Eq. (6)) involves non-linear parameters; consequently, the non-linear least-square method, which is much more complex than the linear least-square method, is needed to determine the optimum values of parameters. Though Eq. (6) is simple, yet is applicable to the entire domain of effective precursor concentrations, and may be applicable approximately to any urban air mass. Therefore, in present work, Eq. (6) is employed to predict ozone concentration based on its precursors in ambient air of Mashhad city.

2.2. Genetic algorithm

Individuals are known as chromosomes[20] in evolution language and every variable in each individual is named as the gene. An ordinary GA that is used for optimization purposes includes following steps:

(1) Initial population is randomly generated in the possible space and design variable domain as defined chromosomes. Simplicity of Gaussian uniform distribution is the reason that it is used as a usual method for creating the initial population. Likewise, the population size should be determined by user[21].

(2) Each chromosome should be represented in binary codes (0 and 1) or float numbers. However, the more natural representations are more efficient and generate better candidates for problem solutions[6], but the kind of representation depends on problem and it is totally case dependent. For the float representation, each variable (x_i) belongs to each gene of chromosome, but in binary form, each variable (x_i) is represented by several genes dependant on required accuracy and domain of each variable. Suppose the required accuracy is S and x_i belongs to $[a_i, b_i]$ so m_i, the number of genes for each variable, would be calculated by the following formula:

$$2^{m_i - 1} < (b_i - a_i) \times 10^S \leq 2^{m_i - 1} \qquad (7)$$

(3) To the point of stochastic view, the fitness function, which evaluates each individual in population set, is independent of the GA operations while depending on minimization, and maximization problem could map the population into specified form[22].

(4) A proper selection method should be picked up among different types of selection procedure for the GA, which is classified as roulette wheel selection, scaling techniques, tournament and ranking selections[6]. Using the roulette wheel after computing a cumulative probability (q_i) for each individual (v_i), random numbers from the domain [0, 1] are generated. Since each random number is located between two consequent cumulative probabilities, the greater one is selected for new population[8]. Unlike the roulette wheel, the tournament selection is used for minimizing. This kind of selection randomly takes two chromosomes from the population, selects the better one, and sends into the sampling pool. This method repeats till the completion of the population size of individuals in sampling pool[23]. The stochastic uniform selection function, as scaling technique, lays out a line in which each parent corresponds to a section of the length of line proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size. However, in all selection methods one or several individuals with the best fitness value(s) should be kept as elitism(s) and should be copied without change in next generation.

(5) Crossover and mutation are crucial parts of genetic operations[24]. By using crossover probability (p_c), two individuals as parents for mating pool are selected and their two new children are produced while one chromosome randomly by mutation probability (p_m) undergoes mutation to produce new one. For the binary system, the uniform mutation and single or two-point crossover are defined. Furthermore, for the float representation, the arithmetic crossover, scattered crossover, single or two-point crossover, intermediate crossover, and heuristic crossover with uniform mutation, non-uniform mutation, multi-non-uniform mutation, and boundary mutation are also used[6].

(6) There are several approaches to stop the GA, the most popular one is the maximum number of generations. Other termination criteria are improvement of solutions from one generation to the next, fitness value and limitation of time.

2.3 Genetic programming

Similar to the GA, the GP is based on simple rules that imitate biological evolution. Combining basis functions, inputs and constants creates an initial model population, whose complexity is controlled by the user. The models are structured in a tree-like fashion, with basic functions linking nodes of inputs, as in the example shown in **Fig. 1**, where the tree-structure for the model: $y = (c_1 \times u_1 + u_2) \times u_3$ is presented, where y is the dependent variable, u_1, u_2 and u_3 are independent variables, and c_1 is a constant. Note that this example also illustrates the organization of the tree in terms of its root and the basic function at the highest level, which in this case is the multiplier function that multiplies u_3 with the remaining elements of the tree. It is noted that the basic functions can be those requiring two arguments, such as '+' and '×' as in the example, or those with only one (e.g., exp $(\cdot)$ or $\sqrt{(\cdot)}$). Each individual model in the population is then fitted to the empirical data using nonlinear regression, and then graded according to how well it matches the data[9]
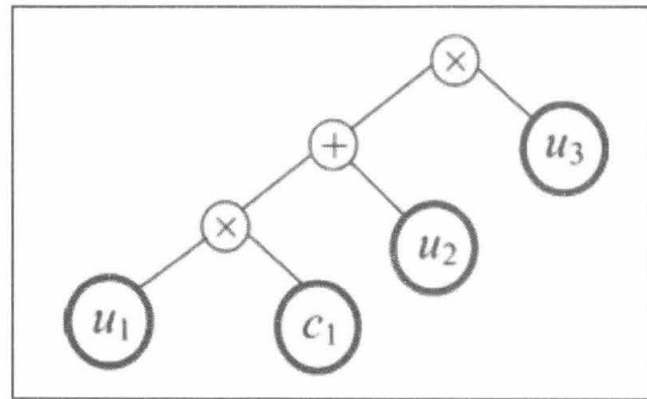

Fig. 1 : Tree-structure for the model (original GP structure): $(c_1 \times u_1 + u_2 \times u_3)$[9, 25]

Regarding **Fig. 2**, which illustrates the flow diagram for a generic GP, in each generation (iteration) of the algorithm, relatively successful individuals are chosen as "parents" for the next generation and form a reproduction pool. A new generation of solutions evolves, using one of three possible operators: crossover, mutation and permutation. In crossover, two individuals from the pool are chosen, their tree structures are divided at a randomly selected crossover point, and the resulting sub-trees are recombined to form two new individuals. In mutation, a random change is performed on a chosen individual by substitution—this can be a functional group, an input variable or a constant. In contrast, branches of a chosen individual are randomly switched in permutation. The parameters for each new individual in the new generation are determined by nonlinear regression, the models are then graded by fitness as before, and the procedure is repeated until a stopping criterion is attained. In most cases, as in this application, the population size, n_{Pop}, and the total number of

generations, n_{Gen}, are decided in advance. Other tuning parameters that need to be fixed by the user when using GPs are p_c, p_m and p_p, the crossover, mutation and permutation probabilities. Additional parameters adopted, are p_i, the probability of a constant to mutate to an input, and for the random population initialization step, N_{tree}, the maximum number of sub-trees in an initial model, and p_{tree}, the probability of creating a sub-tree[9].
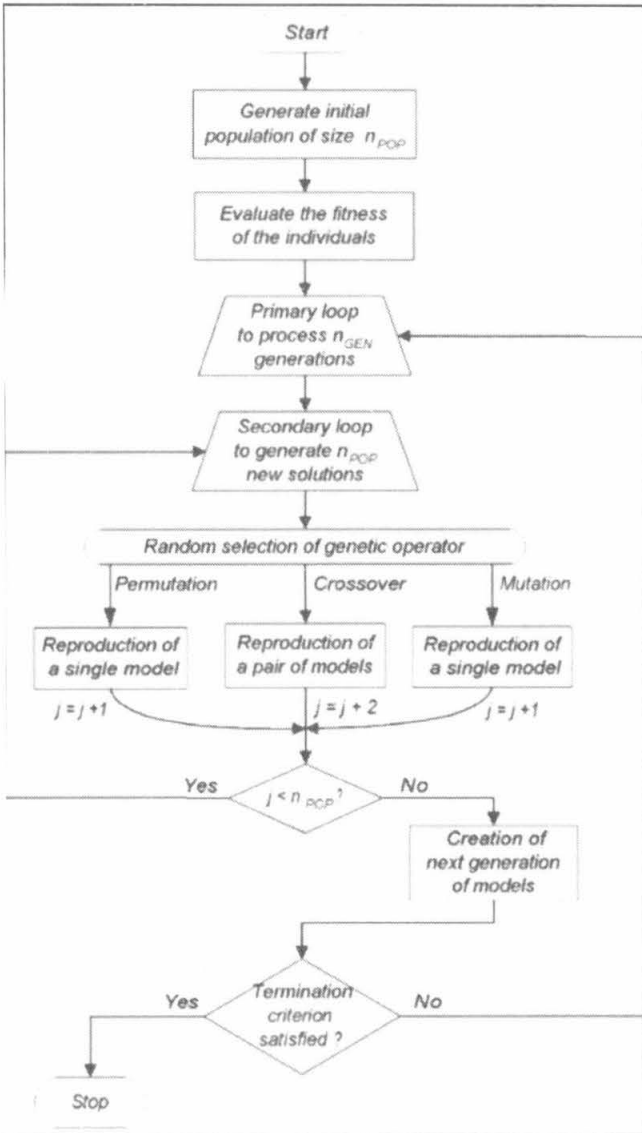

Fig. 2 : Generic GP scheme[9]

2.4 Simulation

The training data obtained from Mashhad city ambient air stations were used to reconstruct the true underlying surfaces representing the dependency of ozone precursors' concentrations and average temperature to ozone concentration using various techniques.

2.4.1 Semi-empirical models regression

The mentioned Eq. (6) in introduction section was used to fit best surfaces based on the coefficient of determination, square root of correlation coefficient, by homemade non-linear regression software. Using this software tried to find the best model parameters which are able to regenerate ozone concentration with an acceptable degree of consistency in comparison with experimental data.

2.4.2 GA

Using MATLAB software, a homemade computer program based on GA was written for optimizing proposed model (Eq. (6)). The objective function, through GA literature fitness function, was the standard square error which should be minimized:

$$E = \sum_{i=1}^{n}\left(L_i - L_{eq,i}\right)^2 \qquad (8)$$

where L_i and $L_{eq,i}$ are measured and calculated ozone concentration at time i, respectively.

Five parameters of model, a, b, c, d and γ, have to be determined by designed GA program which reads the measured ozone precursors' concentration and average run time temperature data for each run time from a spreadsheet provided for the program.

In the genetic algorithm, a population of points with each point representing a set of design parameters has to be formed in a given range. Adjusting this range is not a well-defined process. However, it is not erroneous to estimate these parameters around the experimental values within a certain range.

Fig. 3 presents the flow chart of used computer program and the procedure is as follow:

- 1: Measured data are inputted.

- 2: The maximum number of generations was specified ($N_{Gen,max}$ = 8000). It was observed that for generation greater than 8000 there was no improvement in minimization procedure.

- 3: The parameters' range was determined as following:

  $a_k \in [-2, 0]$, $b_k \in [1, 2.5]$, $c_k \in [0, 1]$, $d_k \in [4000, 5000]$, and $\gamma_k \in [0, 2]$,

---
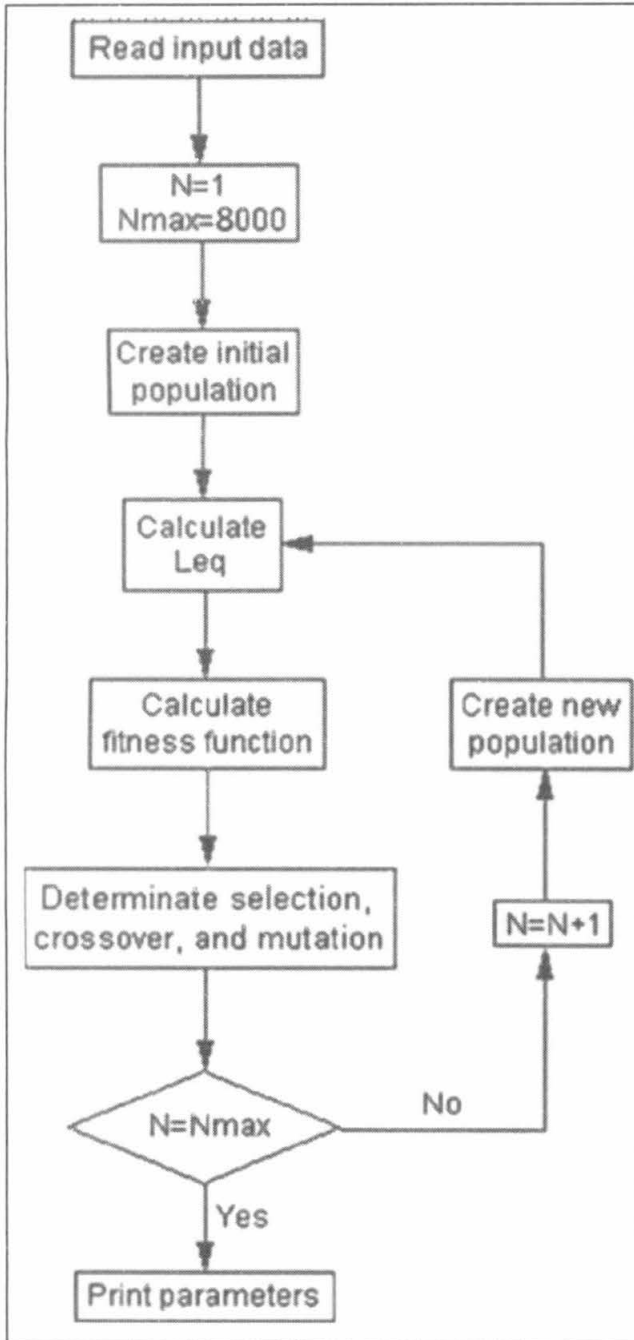
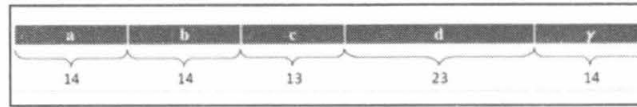Fig. 3 : Flow chart of used GA program

Fig. 4 : Sample of a chromosome (point) from population

- 4: Ozone concentration, L_{eq}, was computed via Eq. (6) fo each chromosome

- 5: Fitness function was evaluated using Eq. (8).

- 6: The GA operations including selection, crossover, an mutation were applied. Five elites were chosen from eac population and added to the new one without an transformations or changes. The crossover probability pc, and mutation probability, pm, were chosen as 0.6 and 0.20, respectively. It notes the values of pc, pm an elites number are determined empirically.

- 7: Finally termination criteria, number of curren generations, were checked. Using aforementione process, the following expression was attained:

$(O_3) = 0.1044 +0.4057 (NO_x)^{1/2}[1-exp(-1.4098R^{2.0150})]exp[$
$4821(1/T-1/T_{av})]$ (9)

2.4.3 GP

The GP approach was implemented in MATLAI (MathWorks). The program was run under a Pentium dual core processor with 2.50GHz and 4GB of random acces memory (RAM).

The parameters of the GP approach were set a follows: population size n_{Pop}=30, and the maximum number c generation n_{Gen}=150. The tuning parameters were fixed a P_c=0.55, P_m=0.25 and P_p= 0.1 the crossover, mutation an permutation probabilities, respectively. The maximum numbe of sub-trees in an initial model, N_{tree}, adopted for this approac was 5, and also P_{tree}, the probability of creating a sub-tree wa assumed 0.5.

The main challenge is to define the qualities of good model and then quantify them into a mathematica expression. In this study, Eq. (10), which is almost similar t the equation considered by McKay et al.[26], was considered a the fitness function for minimization proposes[9]:

$$f_i = \frac{\alpha F_M + (1-\alpha)F_P}{1 + exp(\gamma[n_b - (M_c - \beta)])} \qquad (10)$$

and the initial population with respect to the parameters of a_k, b_k, c_k, d_k and $_k$ (where k represents each chromosome) was randomly generated with 1000 strings (chromosomes). The accuracy value was considered as S = 13, so based on Eq. (7) the number of bites (genes) for the a_k was calculated as 14 and for the other parameters it was 14, 13, 23 and 14, respectively. Consequently, each chromosome consisted of 78 bites according to **Fig. 4.**

where F_M and F_p are the model and prediction fitness, calculated by:

$$F_{M,P} = \frac{\delta}{\delta + SSE} \tag{11}$$

where the sum of squares of errors (SSE) is computed using the modeling data set for F_M and using the validation data set for F_p. In both cases, δ is the standard deviation of the data records, and the fitness values therefore vary from zero (when the SSE value is infinite) to unity (when SSE = 0). In Eq. (10), α is the fraction of the fitness value computed on the basis of modeling performance, is a constant that determines the slope of the complexity constraint sigmoid, n_b is the complexity of the inspected model, which is equal to the number of its branches, M_c the best model complexity of the previous generation, and is updated in each generation, and is a constant responsible of the ability of the algorithm to accept a solution of greater complexity than M_c.

3. Results and discussion

The accuracy of prediction is reflected by some statistical parameters i.e. normalized bias (NB), standard squared error (SSE), mean squared error (MSE), and root mean squared error (RMSE), which are described using equations (12) to (15), respectively. In these relations, L can be ozone concentration. The results for all presented models are compared in Table 1. Passing the fitted surfaces through all training data points is a favor, in present work, as data points has no considerable noises so statistical parameters can be used as termination criteria for desired models.

$$NB = \sum_i^n \frac{(L_{model.i} - L_{exp.i}) / L_{exp.i}}{n} \tag{12}$$

$$SSE = \sum_i^n (L_{model.i} - L_{exp.i})^2 \tag{13}$$

$$MSE = \frac{\sum_i^n (L_{model.i} - L_{exp.i})^2}{n} \tag{14}$$

$$RMSE = \sqrt{\frac{\sum_i^n (L_{model.i} - L_{exp.i})^2}{n}} \tag{15}$$

Table 1 : Results of statistical parameters for presented models

Method	NB	SSE	MSE	RMSE
Employed semi-empirical model	58.3545	0.0889	5.0810e-004	0.0225
GA based on Eq. (6)	27.0941	0.0590	3.3732e-004	0.0184
GP	14.8537	0.0435	2.4878e-004	0.0158

Considering **Table 1**, in this specific case with regard to NB values both employed semi-empirical and GA models highly over predict ambient ozone concentration though GP almost closely predicts those based on ozone precursors and average temperature. The values of SSE, MSE and RMSE show the results of GP have some insignificant errors that can be ignored. It should be noted that although the obtained GA model is not as accurate as the GP one, it presents better results to regenerate the experimental data than the model obtained with respect to semi-empirical Eq. (6).

Figs. 5 and 6 depict the best recall and generalization performances of both employed semi-empirical and GA models, respectively. As it can be seen in those figures, the predicted data points by both employed semi-empirical and GA models neither fit to experimental data with an acceptable tolerance nor provide reliable results for any ranges of NO_x/NMHC ratio. But it is clearly obvious that in spite of the results obtained from the best model based on semi-empirical Eq. (6), they do not even follow the trend of experimental data; the regenerated data points with the best GA model reproduce that trend closely to the best.

Fig. 7 (a) illustrates the best (optimal) recall performances of trained GP model. Evidently, the hyper-surface generated by GP approach almost passes through most of training data points. The corresponding generalization performance of this model as shown in **Fig. 7 (b)** shows the practiced algorithm provides realistic values especially for the lower ranges of NO_x/NMHC ratio.

It is noted that at large R (or NMHC rich regime), the semi-empirical models do not take into account NO_x sinks by some NMHC species, such as aromatics[18]. But, as genetic programming basically considers only the given data it is able to cover all the interactions which are neglected by semi-empirical models, the derived GP model has better performance in this case and do that as well.
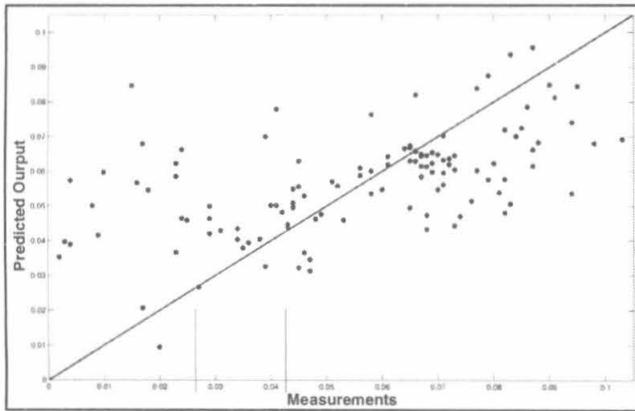
Fig. 5 (a) : Recall performances of the best model based on
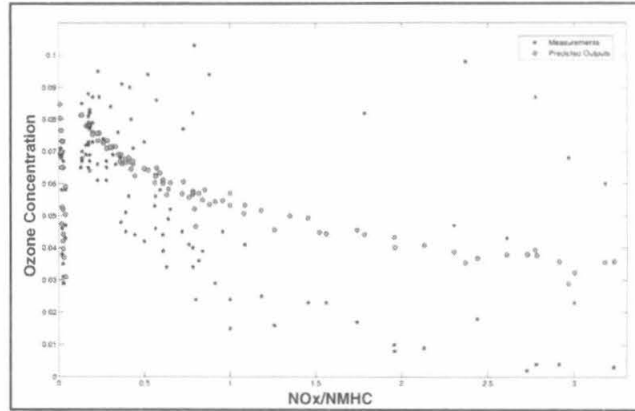employed semi-empirical Eq. (6)

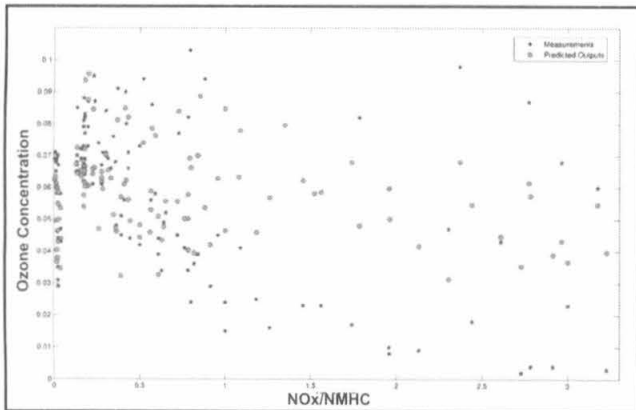Fig. 6 (b) : Generalization performances of the best GA
model

Fig. 5 (b) : Generalization performances of the best model
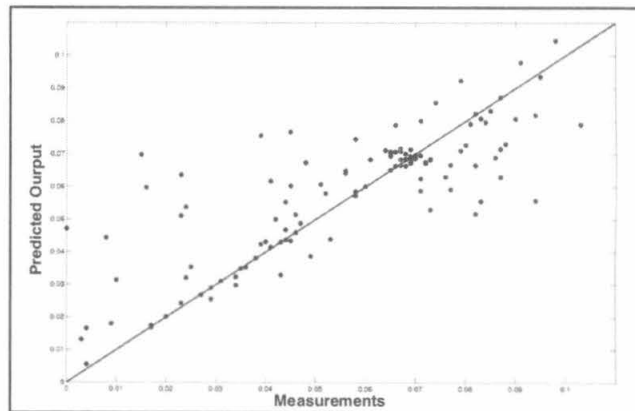based on employed semi-empirical Eq. (6)

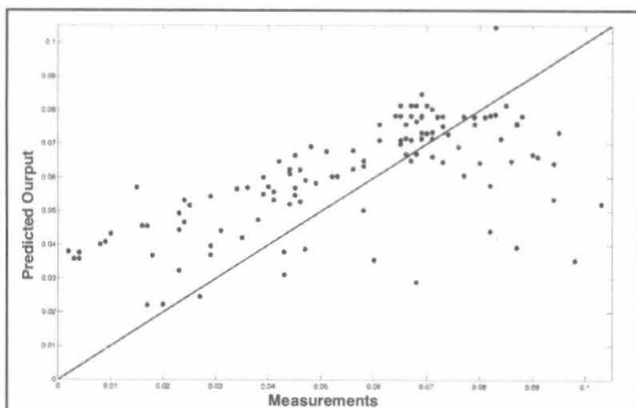Fig. 7 (a) : Recall performances of the best GP model

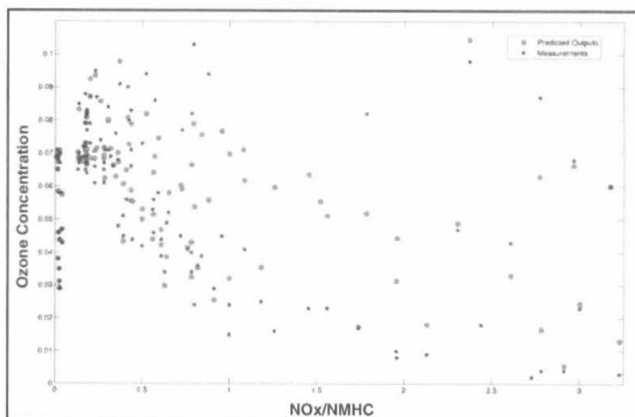Fig. 6 (a) : Recall performances of the best
GA model

Fig. 7 (b) : Generalization performances of the best GP mode

4. Conclusions

The generalization performances of the in-hous
genetic programming model and the model based on geneti
algorithms were compared together and with the prediction

of optimal conventional semi-empirical models. The simulation results indicate that both genetic programming and genetic algorithms provide better predictions but the genetic programming ones are more reliable and accurate. The semi-empirical models which were presented to predict ozone concentrations in any ambient air conditions are not comprehensive enough to perform their task with different set of experimental data very well.

Acknowledgement

References

1. Gray G J, Li Y, Murray-Smith D J and Sharman K C, Structural system identification using genetic programming and a block diagram oriented simulation tool, *Electron Lett*, 32, 1422–1424 (1996).

2. Conniff R, Deadly silk, *Natl Geogr*, 200(2), 30–45 (2001).

3. Li P C, Song K P and Shang F H, Double chains quantum genetic algorithm with application to neuro-fuzzy controller design, *Adv Eng Softw*, 42, 875–886 (2011).

4. Haupt R L and Haupt S E, *Practical genetic algorithms* (Wiley-Interscience Publication, New York), 1998.

5. Koza J R, *Genetic programming* (MIT Press, Cambridge), 1992.

6. Michalewicz Z, *Genetic algorithms + data structures = evolution programs, AI series* (Springer-Verlag, New York), 1996.

7. Shahsavar M, Akhavan Niaki S T and Najafi A A, An efficient genetic algorithm to maximize net present value of project payments under inflation and bonus–penalty policy in resource investment problem. *Adv Eng Softw*, 41, 1023–1030 (2010).

8. Sivanandam S N and Deepa S N, *Introduction to genetic algorithms* (Springer-Verlag, Berlin Heidelberg), 2008.

9. Grosman B and Lewin D R, Adaptive genetic programming for steady-state process modeling, *Computers Chem Eng*, 28, 2779-2790 (2004).

10. Grosman B, *Nonlinear system modeling using genetic programming*, M.Sc. Thesis, Technion, 2001.

11. *Air quality criteria for ozone and other photochemical oxidants*, US. Environmental Protection Agency: Research Triangle Park, NC, 27711, Vol. 2, EPA/600/8-84/020bF (1986).

12. Kuntasal G and Chang T Y, Trends and relationships of O3, NOx and HC in the south coast air basin of California, *JAPCA*, 37, 1158-1163 (1987).

13. Sillman S, The relation between ozone, NO_x and hydrocarbons in urban and polluted rural environments, *Atmos Environ*, 33, 1821-1845 (1999).

14. Lawson D R, The southern California air quality study, *J Air Waste Manage Assoc*, 40, 156-165 (1990).

15. Akimoto H, Sakamaki F, Hoshino M, Inoue G and Okuda M, Photochemical ozone formation in propylene-nitrogen oxide-dry air system, *Environ Sci Technol*, 13, 53-58 (1979).

16. Shen C H and Springer G S, Photochemical particulate formation in cyclohexene-nitrogen dioxide-air mixtures, *Atmos Environ*, 11, 683-688 (1977).

17. Kelly N A and Gunst R F, Response of ozone to changes in hydrocarbon and nitrogen oxide concentrations in outdoor smog chambers filled with Los Angeles air, *Atmos Environ*, 24, 2991-3005 (1990).

18. Chang T Y and Rudy S J, Ozone-precursor relationships: a modeling study of semiempirical relationships, *Environ Sci Technol*, 27, 2213-2219 (1993).

19. Press W H, Teukolsky S A, Vetterling W T and Flannery B P, *Numerical recipes* (Cambridge University Press, New York) 2007.

20. Corriveau G, Guilbault R and Tahan A, Genetic algorithms and finite element coupling for mechanical optimization. *Adv Eng Softw*, 41, 422–426 (2010).

21. Pasandideh S H R, Akhavan Niaki S T and Yeganeh J A, A parameter-tuned genetic algorithm for multi-product economic production quantity model with space constraint, discrete delivery orders and shortages, *Adv Eng Softw*, 41, 306–314 (2010).

22. Back T, Fogel D B and Michalewicz Z, *Handbook of evolutionary computation* (University Oxford Press, New York) 1997.

23. Shopova E G and Vaklieva-Bancheva N G, BASIC-A genetic algorithm for engineering problems solution, *Computers Chem Eng*, 30, 1293–1309 (2006).

24. Sahman M A, Çunkas M, Inal S, Inal F, Coskun B and Taskiran U, Cost optimization of feed mixes by genetic algorithms. *Adv Eng Softw*, 40, 965–974 (2009).

25. Grosman B and Lewin D R, Automated nonlinear model predictive control using genetic programming, *Computers Chem Eng*, 26, 631–640 (2002).

26. McKay B, Willis M and Barton G, Steady-state modeling of chemical process systems using genetic programming, *Computers Chem Eng*, 21(9), 981–996 (1997).