Prediction of Aluminum Porosity Using Approximate Functions Generated by Evolutionary Algorithm

Natnapat Gaviphatt*[1], Prabhas Chongstitvatana*[2], Chedtha Puncreobutr[#3]
*Department of Computer Engineering
[#]Department of Metallurgical Engineering
Chulalongkorn University, Phayathai Road, Pathumwan, 10330
Bangkok, Thailand
[1]Natnapat.g@student.chula.ac.th, [2]prabhas.c@chula.ac.th, [3]chedtha.p@chula.ac.th

***Abstract* — Aluminum alloy is one of the popular materials due to its abundant supply and its good properties. However, its casting process has one undesirable effect - porosity. There are many attempts to predict its formation in order to avoid it. In this research, a way to predict total porosity percent from the initial chemical compositions and cooling rate after casting is proposed. The prediction is in a form of prediction function which utilizes the combination of genetic algorithm and differential evolution to generate the predicting formulae with datasets from other researches. The result functions achieve satisfying accuracy showing that they are capable of the good prediction.**

Keywords—genetic algorithm; differential evolution; aluminum; casting; porosity

I. INTRODUCTION

Aluminum alloys nowadays are very important materials. They are almost everywhere around us. Thanks to the abundance of aluminum on earth and their durability, lightness and corrosion resistance, they have become greatly preferred for areas of transportation and structural materials [1]. However, in forming process, especially casting which is appropriate for complex shapes, porosity arises and undesirably degrades mechanical properties which waste the cost and time in the industrial process.

As an important problem, there are many researches involving prediction of porosity formation. In the beginning era, they are in the form of analytical models [2, 3] which were later developed to thermodynamic calculations and fluid flow simulations[4], extinguishing most of the macro porosity problems in the process. However, not a single existing model for micro porosity could overcome limitations and completely surpass industrial problems in every case [5]. After development of artificial neural networks had become mature, some researches try it for the prediction and gained satisfying results [6, 7]. Nevertheless, the error should still be able to reduce further.

This research aims to use function approximation technique, which requires a target function, combined with polynomial approximation to synthesize the porosity amount prediction function. In order to achieve this, the target function as well as datasets is inferred from other researches. Genetic Algorithm is appointed to search for polynomial forms while Differential Evolution provides coefficients for each polynomial term. In the end, accuracies are compared with models acquired from neural networks.

II. THEORETICAL BACKGROUNDS

A. Porosity formation

There are two main causes of porosity in alloy casting especially for aluminum [8]. The first type is gas porosity which happens from loss of ability of solvent to dissolve gas, hydrogen in case aluminum. To be precise, hydrogen gas is released as the alloys liquid cools down and solidifies. Then, with more cooling rate, there would be less time for hydrogen gas to form as pores resulting in less porosity. The second one is shrinkage porosity which arises from significant contraction of chemical bonds when solidify. In other words, when liquid alloys pass through solidification state, shrinkage occurs and with the lack of feeding and pressure, empty spaces are inevitable. This means that with higher cooling rate, porosity amount would increase because solidification occurs more rapidly forming more dendrites to resist the flow which make it more difficult to fill empty spaces.

B. Genetic Algorithm (GA)

The most widely known type of evolutionary algorithm is GA. Evolutionary algorithm is a group of search and optimization algorithms that mimics the process of Darwinian evolution. At first, for any problems, relevant solution must be transformed into an appropriate form (genotype) which is called *'chromosome encoding'* or *'chromosome representation'*. The first generation of solutions is, then, randomly generated as an initial population. New generations of solutions would be iteratively generated from the previous one either from *recombination* or *mutation* or even both. Given a fitting function to be optimized, natural selection could be applied (survival of the fittest) which causes a rise in the fitness of each generation. Genetic algorithm, as being the most basic one, strongly inherits those ideas. Even though it was originally created as a means of studying adaptive behaviour, it is now a potent optimization method [9].

1) Representations

Various *chromosome representations* exist in GA. The simplest one is binary representation consisting of a string of binary digits as its genotype. Other similar representations are integer and real-value that use integer numbers and floating-point numbers instead of digits for their genotype respectively. Some other representations also exist such as tree representation and permutation representation which are even more complex. Representation selection is one of the most important steps for implementation of GA.

2) Recombination

They are actually many ways of reproducing a new solution from the former generation's characteristics contained within two or more parents. Recombination of GA varies with representation. One of a basic technique is *uniform crossover*. The process is that an offspring chromosome is randomly selected from parents independently for each position. As for complex representations, they need more complicated recombination techniques.

3) Mutation

It is a generic name given to any methods of modifying an existing individual into a new one by applying some randomized change. Similar to the recombination, mutation varies with representation. *Uniform mutation* is the easiest one in case of the three basic representations. In its process, with probability p_m, often referred to as mutation rate, a new value is chosen randomly from a set of permissible values in each position.

4) Selection

Selection is a mechanism to select a group of survivors to be a new generation. The selection could pick survivors from both μ parents and λ offspring ((μ + λ) selection). While selection only from λ offspring ((μ, λ) selection) is an alternative way to avoid premature convergence – the situation that searching converges too early before finding the global optima. Algorithm to choose survivors is also one important topic for consideration. A good one is elitist which allows only the μ best individuals to survive as the new generation.

C. Differential Evolution (DE)

DE holds ideas similar to those of GA. Though it is developed to be reliable and versatile function optimizer, it is also easy to use. It is designed to efficiently move around continuous search-space making it suitable for numerical optimization problem. Representation for DE is real-value which is now recognized as vector [10].

Mutation, recombination and survivor selection are all changed to DE style and need to be operated in order. In each round, every individual in the population would be operated and firstly considered as target vectors. In differential mutation process, other 3 vectors r_1, r_2 and r_3 are randomly picked from the population and combined with simple arithmetic operator to create mutant vector v_i as shown in (1) when F is a scale factor, F ϵ (0,1+). But with another mutation strategy such as DE/rand/1 either-or-algorithm, mutant vector combination uses (2) instead.

$$v_{i,g} = r_{1,i,g} + F \cdot (r_{2,i,g} - r_{3,i,g}) \tag{1}$$

$$v_{i,g} = r_{1,i,g} + (r_{2,i,g} + r_{3,i,g} - 2r_{1,i,g}) \cdot \frac{(F+1)}{2} \tag{2}$$

Recombination is operated after mutation. Technically it is similar to GA's uniform crossover. The crossover builds trial vector u_i out of parameter values that have been taken from target vector x_i and mutant vector:

$$u_{i,j} = \begin{cases} v_{i,j} & if\ (rand_j(0,1) \le Cr \text{ or } j = j_{rand}) \\ x_{i,j} & otherwise. \end{cases} \tag{3}$$

Crossover probability, Cr ϵ [0, 1], is a user-defined value to specify the fraction of new and old parameter values.

The last part of iteration is the selection. It is a step to compare whether newly produced trial vectors are better than their target vectors or not. Only the trial vectors that beat their counterparts are allowed to replace the position and stay as new candidate solutions.

III. Methodology

A. Selecting a Template Formula

According to the explanation of porosity formation, it could be inferred that there are two trends of formation focusing cooling rate. The first one is decreased if the rate goes higher which resemble exponential decay the most. While the other one is increased along with the rate. Even though exponential growth or power functions are able to explain the inclination, their infinite growing property is unacceptable, making the logistic function to come in their stead. Dependence between the two types of porosity is still unknown, so it is only allowed to hypothesize that they are independent of one another. With that, the total porosity is summation of them as in (4).

$$p\% = p_{hydrogen} + p_{shrinkage} \tag{4}$$

When p% is total porosity percent. If other hypotheses previously made are also applied then the equation (5) could be the result.

$$p\% = \frac{P_1}{e^{P_2 + P_3 R}} + \frac{P_4}{1 + e^{P_5 - P_6 R}} \tag{5}$$

When P_i are polynomials that act as graph magnitude adjusters, translators and stretchers and R is the cooling rate.

B. GA and DE Combination

To generate proper the polynomials which fit the equation to experimental data, the datasets of independent variables

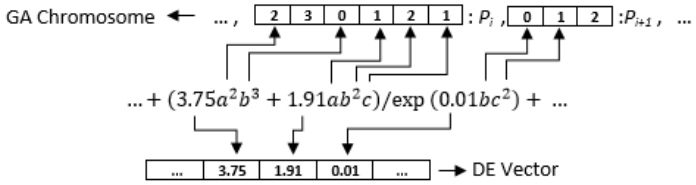


Fig. 1. GA-DE representations model

consist of initial chemical composition and cooling rate are used. GA is used in conjunction with DE to find parameters in the equation (5). GA is responsible for generation of terms and power of each variable, while DE is for finding of the appropriate parameters of each term.

Actually, a normal GA representation and recombination could handle this problem. However, to enable variable length property of polynomials, a special form is necessary. For representations, powers of each variable are encoded into chromosomes of integers separately for each polynomial. In other words, if there are six polynomials in an equation, there are six chromosomes. As for recombination a slightly more complicated technique is applied but it would be explained later. About DE, parameters of each polynomial term are normally serialized into a vector as shown in Fig. 1.

The overall work flow of GA-DE combination method is shown in Fig. 2. It is mainly focused in GA and employs DE only before the step of survivor selection to evaluate the error values. The flow could be summarized as follow:

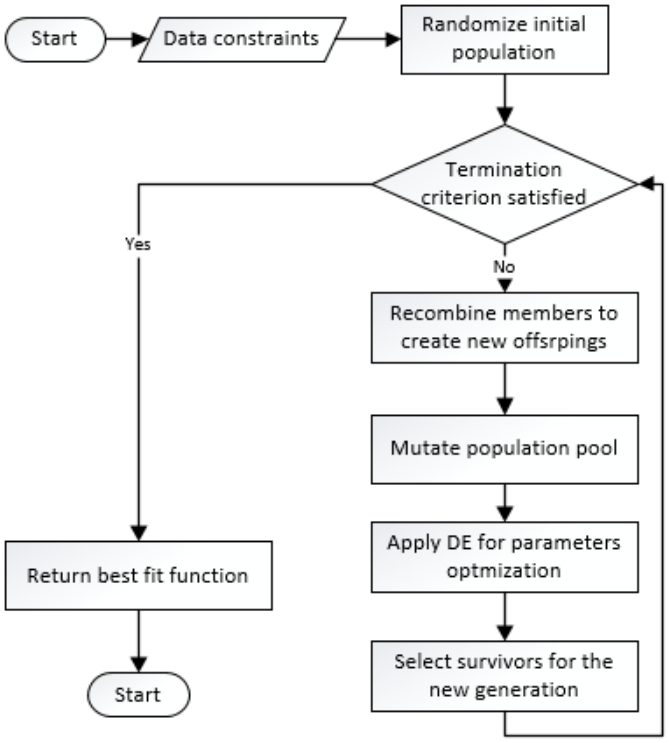


Fig. 2. Flow chart of GA-DE method

- Specify the data and various configuration constraints.
- Randomize initial members to be the first generation of populations with number of population (μ) 250.
- Check if the termination criterions are satisfied or not. If they are satisfied, then return the best fit formula and stop the operation but if they are still not satisfied, repeat the following steps until they are.
- Produce new offspring by recombine members in the last generation population pool with number of offspring (λ) 75.

```
Recombination

n : number of parents to crossover
m : number of polynomials
P1,P2,...,Pn : parents of crossing over
c1,c2,...,cm : maximum length of each polynomial

P = initialize new Equation;
for i = 1 to m{
        P(i).addTermsFrom(P1(i)..Pn(i));
        P(i).removeDuplicatedTerms();
        P(i).randomRemoveTermsUntil(ci);
        for each (Term t in P(i)){
                if(random(0,1) < Dr)
                        P(i).remove(t);
        }
```

- Apply uniform mutation to offspring with mutation rate (p_m) 0.3
- Employ DE acquire the relevant coefficients for every single new equations with the configuration stated in TABLE. I.
- Select the survivors by evaluating the error of every members

TABLE I. CONFIGURATION OF DE

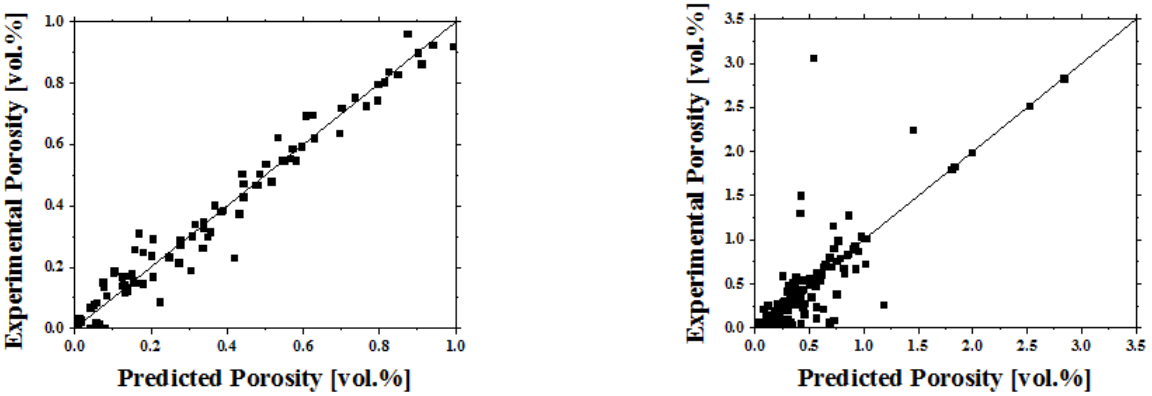
Configuration	Value
Population (μ)	200
Scale Factor (F)	0.85
Crossover Probability (Cr)	0.75
Strategy	DE/rand/1 either-or-algorithm

C. GA's recombination

Recombination of GA needs to be modified in order to achieve variable length of chromosomes. The first constraint is that a chromosome could only crossover with another one of the same polynomial position because a term from a position should not disturb terms from other position. Another constraint is the maximum length of each polynomial position. Although the chromosome's length could be variable, it should not grow to infinite. This modification only changes the crossing over between parents, so the ways of parent selection

could be the same as normal GA. The pseudo code for the crossing over is shown here.

	r	0.8187	0.2053



(a) (b)

Fig. 3. Predicted porosity plotted with experimental value from (a) dataset 1 and (b) dataset 1+2

The important parts of the operation are the part that removes duplicated terms and the part that randomly removes terms at the rate of *Dr* – deleting rate It should be the value between (0, 1). These parts make the polynomial variable in length and greatly increase GA's performance.

D. Datasets and Evaluation

Experimental data of porosity (vol. %) in this research are gathered from previous researches. Dataset 1 is from [11] which gives 96 samples with initial amount of Si, Fe, Cu, Mg, Mn, Sr, Ti (wt. %) and cooling rate (°C) as independent variables. Dataset 2 is from [12] and gives other 48 samples with the same independent variables as datasets 1 plus initial amount of hydrogen and other elements which were omitted to allow it to merge with dataset 1.

The first dataset would be fed to GA-DE model followed by dataset 1+2 to be more challenge since the merged dataset must possess more complexity.

IV. Results and Discussion

Prediction results from formulae generated by GA-DE model are shown in Fig. 3. It is demonstrated that prediction values over dataset 1 are promisingly accurate. This is because the data is from only a single source. In contrast, the prediction over the mixed dataset provides worse performance. There are groups of data that prediction gives good results as well as the bad ones. The bad results are the results from incompatibility of datasets.

TABLE II. COMPARISON OF ERRORS

Data	Error	Proposed Model (vol.%)	Neural Networks (vol.%)
Dataset 1	rms	0.0502	0.0643
	mae	0.0353	0.0472
	r	0.9941	0.9765
Dataset 1+2	rms	0.3029	0.6591
	mae	0.1386	0.3871

After GA-DE finish its process, the resulted functions are then compared with the models acquired from performing ten-fold cross validation from neural networks with 5 – 15 hidden nodes that give the best fit. The results are expressed in TABLE II. Errors provided, consisting of root mean square (rms), mean absolute error (mae) and correlation coefficient (r), are normal tools to evaluate the performances of prediction models. The proposed model proved better in every field

V. Conclusion

This research proposed a method to predict the total porosity formation percent from aluminum alloy casting process. The method utilizes the traditional function approximation in conjunction with polynomials approximation by GA and DE to fill unknown positions. The accuracies of predicting functions are satisfying. They beat models from neural networks, which were designed to learn and predict complicated data, guaranteeing performance of the generated formulae. Moreover, the predicting functions can be used for factors' influence analysis. This is simpler than using models from neural networks. These benefits could obviously improve industrial process by reducing chances of wasting product.

References

[1] R. N. Lumley, "1 - Introduction to aluminium metallurgy," in *Fundamentals of Aluminium Metallurgy*, R. Lumley, Ed., ed: Woodhead Publishing, 2011, pp. 1-19.

[2] T. S. Piwonka and M. C. Flemings, "Pore Formation in Solidification," in *Transactions of the Metallurgical Society of AIME*, ed: American Institute of Mining, Metallurgical, and Petroleum Engineers, 1966, pp. 1157-1165.

[3] K.-D. Li and E. Chang, "Mechanism of nucleation and growth of hydrogen porosity in solidifying A356 aluminum alloy: an analytical solution," *Acta Materialia*, vol. 52, pp. 219-231, 1/5/ 2004.

[4] D. R. Gunasegaram, D. J. Farnsworth, and T. T. Nguyen, "Identification of critical factors affecting shrinkage porosity in permanent mold casting using numerical simulations based on design of experiments," *Journal of Materials Processing Technology*, vol. 209, pp. 1209-1219, 2/1/ 2009.

[5] P. D. Lee, A. Chirazi, and D. See, "Modeling micro porosity in aluminum alloys: a review," *Journal of Light Metals*, vol. 1, pp. 15-30, 2001.

[6] S. H. M. Anijdan, A. Bahrami, H. R. M. Hosseini, and A. Shafyei, "Using genetic algorithm and artificial neural network analyses to design an Al-Si casting alloy of minimum porosity," *Material and Design*, vol. 27, pp. 605-609, 2006.

[7] A. M. Hassan, A. Alrashdan, M. T. Hayajneh, and A. T. Mayyas, "Prediction of density, porosity and hardness in aluminum–copper-based composite materials using artificial neural network," *Journal of Materials Processing Technology*, vol. 209, pp. 894-899, 2009.

[8] I. Ghosh, S. Das, and N. Chakraborty, "An artificial neural network model to characterize porosity defects during solidification of A356 aluminum alloy," *Neural Computing and Applications*, vol. 25, pp. 653-662, 2014/09/01 2014.

[9] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*: Springer, 2003.

[10] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*: Springer, 2005.

[11] M. Dash and M. Makhlaouf, "Effect of key alloying elements on the feeding characteristics of aluminum–silicon casting alloys," *Journal of Light Metals*, vol. 1, pp. 251-265, 2001.

[12] N. Roy, A. M. Samuel, and F. H. Samuel, "Porosity formation in AI-9 Wt Pct Si-3 Wt Pct Cu alloy systems: Metallographic observations," *Metallurgical and Materials Transactions A*, vol. 27, pp. 415-429, 1996/02/01 1996.