Genetic Programming Approach to Determining Thermal Properties of Lead-free Solder Alloys

Chedtha Puncreobutr[1], Boonrat Lohwongwatana[2], Prabhas Chongstitvattana[1,*]
[1]Department of Computer Engineering, Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok, 10330, Thailand
[2]Department of Metallurgical Engineering, Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok, 10330, Thailand
*Email: Prabhas.C@chula.ac.th

Abstract

In this paper, the model of genetic programming, one of the evolutionary computation methods, was developed to predict melting point and solidification range of solders. The training data set consisted of solidus/liquidus temperatures that were calculated from the calculation of the phase diagrams (CALPHAD) approach. The difference between the training data and predicted data were used as the fitness function of the model via root mean square error method. Accuracy of the model was verified with the experimental results of various compositions of Sn-Ag-Cu-Bi-In solder alloys from the references. The agreement of predicted results and experimental data from literatures can be used to conclude that genetic programming is a powerful algorithm to predict the thermal properties of solder. It also can be used to demonstrate the relationship between the solder alloy compositions and their properties. However, the optimal search performance is also dependent of the parameter setting of the algorithm.

Key Words: Genetic Programming; Lead-free Solder Alloys; Thermal Properties; Modeling

1. Introduction

Lead-based solders, particularly tin-lead solders, have been widely used in electronics industry for a long time. This is primarily due to a combined merit of low cost, adequate melting temperature range, good soldering properties, and proper physical, mechanical properties [1]. However, due to increased concerns regarding the toxicity and environmental impacts of lead, the RoHS directive was installed since 2006. From that year, no drop-in solutions for the banned Pb-Sn solder alloy have been found. To design a new solder alloy, many key performance properties must be considered and optimized, for example, liquidus and solidus temperatures, the reflow-ability, etc. There have been constant efforts to improve the properties of lead-free solder alloys in various aspects. Very often, many researchers have focused on the property improvements which are influenced by minor additions of various alloying elements. Such trial-and-error experiments are costly and time-consuming especially when a wide multitude of multi-component alloy compositions needs to be investigated.

Recently, the basis of natural evolution has been successfully used for solving various engineering problems. With these evolutionary computation methods, it is possible to analyze a huge number of solutions in the feasible region within the short time. Genetic Programming (GP) is one of the evolutionary computation methods that mimic the nature by applying some genetic operators to tree-based solutions and allowing only superior genetic traits to survive. Many fields of research have been investigated [2-4] by using genetic programming to find out the solution.

In the proposed paper, genetic programming was investigated to predict melting temperature and solidification range of solder alloys by learning liquidus and solidus temperature data of various solder compositions that were calculated through CALPHAD method. The predicted results were compared with the experimental results from literature to verify the accuracy of the model. Moreover, the dependent of optimal search performance of the model and the parameter setting of the algorithm was also studied.

2. Genetic Programming

Genetic programming (GP) is an automated methodology inspired by biological evolution to find computer programs that best perform a user-defined task. Genetic programming was first introduced by John Koza [5] to solve genetic algorithms' weakness.

In genetic algorithms, genetic structure is adequate to solve many problems but it is restrictive when size or forms of solution cannot be assessed beforehand. Koza's extension, solutions of which were the program syntax trees, permits explicitly hierarchical variable length. The tree-based solutions of genetic programming can be computer programs, mathematic expressions, state transition rules, etc.

Possible solutions in genetic programming are composed in a recursive manner from a set of function genes ***F*** and the set of terminal genes ***T***. The set of function genes ***F*** can include basic mathematical functions, boolean functions, relation functions, program flow control functions, and functions defined with respect to the problem area studied. The set of terminal genes ***T*** can include numerical constants, logical constants, variables, etc. Terminal genes are, in fact, function genes without arguments.

First, the initial population was obtained from random creation of computer programs (trees) consisting of available set of function gene and set of terminal gene. Each individual population represents a random possible solution in the search space and its genetic structure represents the individual characteristics of each solution. Next step is the fitness measurement of the individuals to indicate the quality of solutions. Then, the genetic operators: reproduction, crossover and mutation, were applied to create the offspring for next generation.

Reproduction gives a higher probability of selection to more successful organisms. They are copied unchanged into the next generation. Crossover is the process that recombines genetic structure of parents in order to generate two new offsprings by exchange fragmented parts of their parents. Figure 1 shows the crossover operation of two tree-based solutions. Mutation is the progress that randomly changes a few genetic structures, subtree, of parents to create one new offspring.

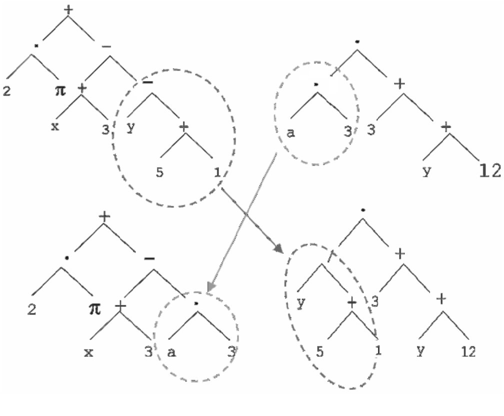

Fig. 1 Crossover in Genetic Programming

The parents and offspring are mixed and go to the selection process. In this process the solution with higher fitness value will have more probability to be selected to the next generation. Number of solutions that were selected in this process was equal to the population size.

After completed all steps of genetic programming which included: (1) initiate population, (2) fitness measurement, (3) genetic modification, (4) selection an iterative repetition of point 2 and 4 follows. After a certain number of generations, the solutions are usually much better adapted to the environment. Therefore, ideally, in each generation the solutions will converge to the best or optimized solutions.

The evolution concept of genetic programming illustrates in the figure 2.

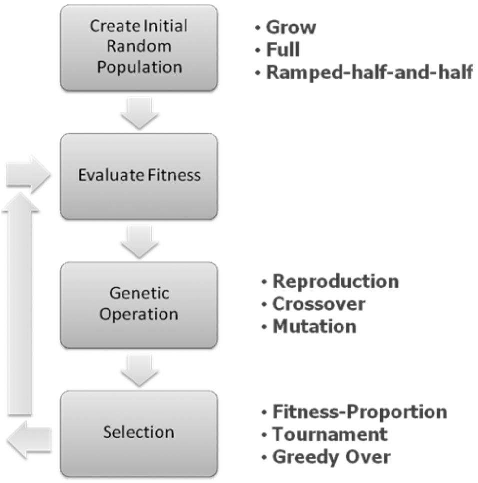

Fig. 2 Genetic Programming Concept

3. Thermodynamic Modeling

Thermodynamic modeling was investigated to calculate phase equilibria in a multi-component system and employed the calculation of the phase diagrams (CALPHAD) approach [6] to describe the basic properties, liquidus / solidus temperature, of candidate solder alloy to replace lead-containing solder.

The current thermodynamic database that used for CALPHAD approach in this research includes the following elements:

Sn-Ag-Cu-Bi-In

Liquidus temperature and solidus temperature play an important role to the reliable of interconnects between metal and solder. Moreover, from these two properties, the thermal characteristics of solder such as melting point and solidification range could be predicted.

4. Methodologies

From the relationship between solder composition and its properties, genetic programming model in this paper was performed with five independent variables. Weighted percent composition of elements in solder: Sn (X_1), Ag (X_2), Cu (X_3), Bi (X_4) and In (X_5) were selected as independent variables in this study.

First, database of liquidus temperature and solidus temperature of 176 solder compositions were calculated via CALPHAD software, Thermo-Calc. From this the training data set for genetic programming was obtained on the basis of five solder composition variables. Each combination of five solder alloys composition, the corresponding liquidus temperature (T_L) and solidus temperature (T_S) were recorded.

Also, the testing data were collected five solder composition variables and corresponding data, liquidus/solidus temperature. While training data were calculated through CALPHAD method, testing data were obtained from the experimental results of literatures. In this paper, testing data were comprised of 32 solder alloy compositions.

The implementation of genetic programming were started from the available function set and terminal set. Function sets ***F*** were basic arithmetical functions (i.e., operations of addition, subtraction, multiplication, and division), square function and natural exponential function. The two latter functions have one argument, whereas the other functions have two arguments each.

Terminal sets $\boldsymbol{T}$ were weighted percent composition of elements in solder: Sn (X_1), Ag (X_2), Cu (X_3), Bi (X_4) and In (X_5). In order to increase genetic diversity of the population the random numbers from the range [0, 1] were added to the set of terminals.

A root mean square error (RMSE) of all sample data for individual in the population was introduced as fitness function. It is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n}}$$

where n is population size, $y \; and \; \hat{y}$ are actual liquidus/solidus temperature and the predicted liquidus/solidus results from the model, respectively.

5. Parameters Setting

The evolutionary process in genetic programming was controlled by many evolutionary parameters. In this research, varying of parameters has been studied for finding a dependency between predicting performance and the parameters. The method to initialize the population in the model was ramped half-by-half. The population size was 20, 50 and 100. Maximum number of generations to be run was 300 and 500. One point crossover with the probability of crossover (Pc) 0.9 and 0.7 were applied. Probability to mutate (Pm) subtree of each individual were 0.1 and 0.25 and the probability to reproduction (Pr) was 0.1 and 0.2. The dynamic growth method was used to preserve the size of solution and the maximum tree depth is 10. The method for selection the individual to next generation was lexicographic tournament selection with tournament size of 4 and 7. Lexicographic selections choose the solution with both best fitness and less complexity (small tree). From this point, ten patterns of parameters were formed and used in this paper; detail of each pattern is shown in Table 1.

Table 1 Parameter setting in genetic programming

Pattern	Pr	Pc	Pm	Tour-Size	PopSize	MaxGen
A	0.1	0.9	0.1	4	20	300
B	0.1	0.9	0.1	4	20	500
C	0.1	0.9	0.1	4	50	300
D	0.1	0.9	0.1	4	50	500
E	0.1	0.9	0.1	4	100	500
F	0.1	0.7	0.25	4	50	500
G	0.1	0.7	0.1	7	50	500
H	0.1	0.9	0.1	7	50	500
I	0.1	0.9	0.25	7	50	500
J	0.2	0.9	0.25	7	50	500

6. Results and Discussion

The genetic programming model was run all patterns (A-J) three times and each run the RMSE of the best solution was collected. The average RMSE value of each pattern indicated the predictive performance of the model; the lowest RMSE is the best parameters setting for genetic programming. Table 2 shows the average RMSE of each pattern.

Table 2 Average RMSE of best solution in each run

Pattern	Average RMSE (3 runs)	
	Liquidus	Solidus
A	5.5849	4.8807
B	5.0805	3.1901
C	6.4114	3.7341
D	4.9033	2.6554
E	4.8800	2.4867
F	4.2284	2.6964
G	4.5077	2.7327
H	5.2669	2.2371
I	5.7009	2.1783
J	4.6046	2.9652

The results showed that the predicted liquidus temperature from pattern F was most fit to the training data with the average RMSE 4.2284. With the parameters setting followed the pattern F the best solution for prediction of liquidus temperature presented by Equation (1).

$$T_L = 1.47908 +2X1-2.02231X2+11X3+2X2X3 +2X4+Log[X3]+Log[16X3^4]+Log[0.094784 +16\ X3^4]^2+Log[X1+4\ X3^2+X4]+Log[-2.02231X2+X3+16X3^4+X4]+Log[X1+X5+(X2+X5)^2]+Log[2X1*Abs[-X2+16X3^4+X5]] \quad (1)$$

This solution was compared to 32 experimental results from references to verify the accuracy of the model. The results demonstrated that the RMSE between experiments and genetic programming was 10.6239. This value showed a little lower than the RMSE between experiments and CALPHAD which was 12.0956.

For the best solution, the deviation of predicted results from the training and testing data set are shown in figures 3-4 and figures 5-6, respectively.

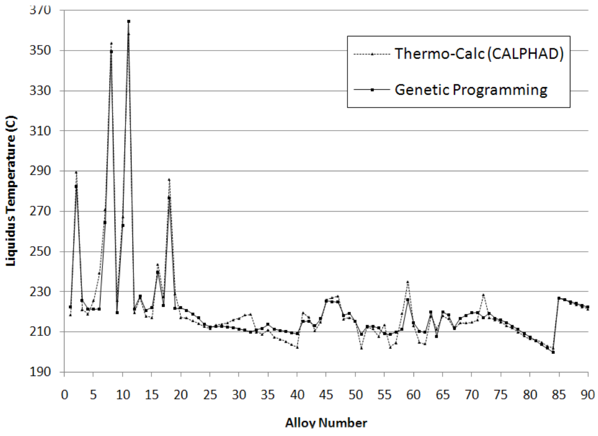

Fig. 3 Liquidus temperature of predicted results and the training data

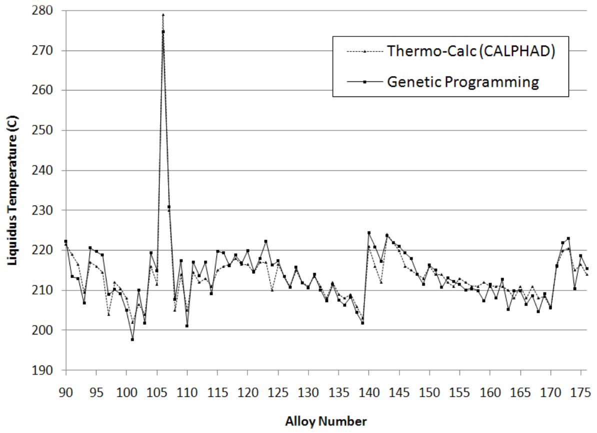

Fig. 4 Liquidus temperature of predicted results and the training data

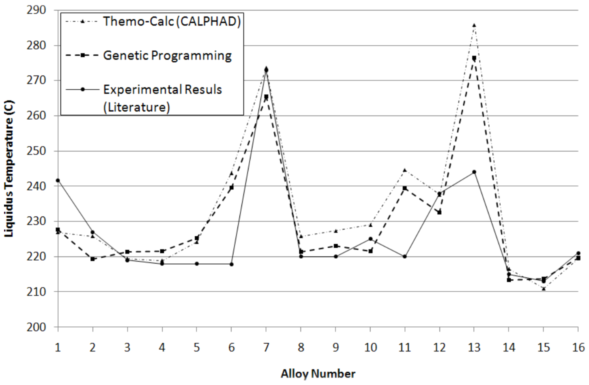

Fig. 5 Liquidus temperature of predicted results and the testing data

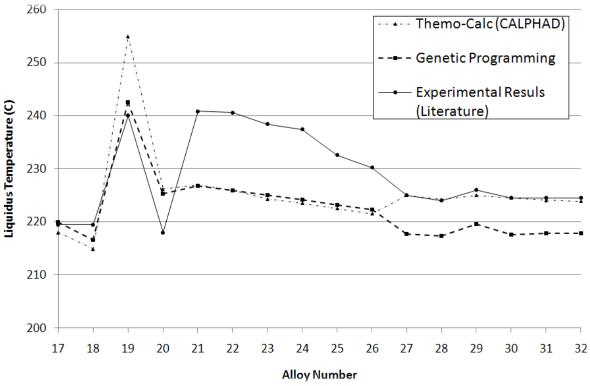

Fig. 6 Liquidus temperature of predicted results and the testing data

While, the pattern F gave the best model for predicting liquidus temperature, the best model to obtain the best solution for predicting solidus temperature was came along with the pattern I (RMSE 2.1783).

Equation 2 shows the best solution for predicting solidus temperature with the parameters setting followed pattern I. Please note that to avoid the error in the model, each division function in this research was set to zero if its denominator was equal to zero.

$$T_S= 0.57736+2X1+X2+X3+\frac{X3}{X2\text{-}X4}+\frac{0.81144}{X4}+\frac{X2}{X4}+\frac{1.76592+X2\text{-}X4}{X4}\text{-}2X4+\frac{0.81144Log[X1\text{-}2X4+X5]}{0.39357+X3^2X5^2}+\frac{X4(0.81144+3X4)(X3\text{-}X5)}{1\text{-}0.420937X4+0.81144X3X4+3X3X4^2} \quad (2)$$

Like liquidus, the predicted results of solidus temperature from the best solution in the model were compared with 32 experimental data from literature. Again, the results show a lower RMSE between experiments and genetic programming (RMSE

4.1696) and a higher RMSE (4.3665) between experiments and CALPHAD.

Based on this best solution, the difference between predicted results and the training data and difference between predicted results and testing data are shown in figures 7-8 and 9-10, respectively.

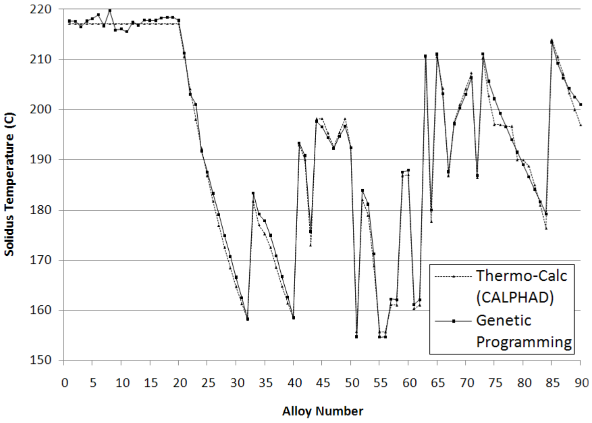

Fig. 7 Solidus temperature of predicted results and the training data

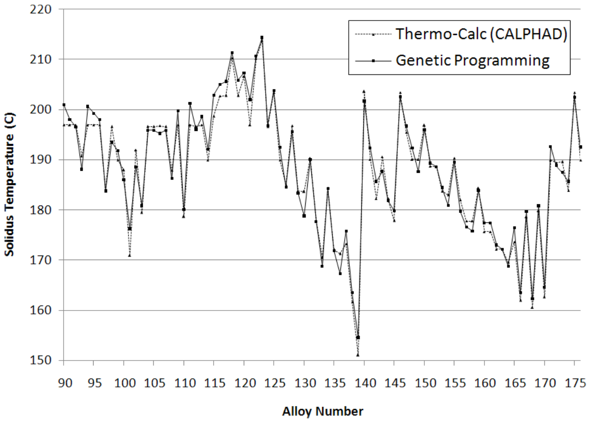

Fig. 8 Solidus temperature of predicted results and the training data

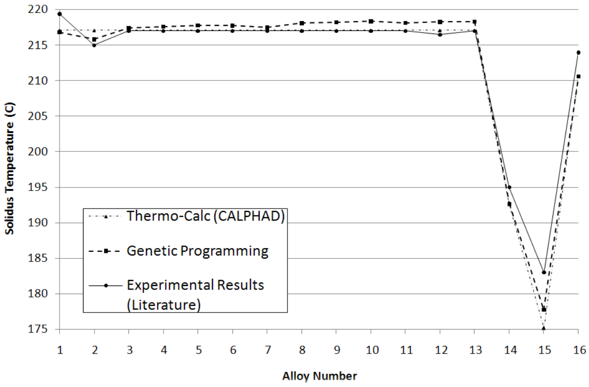

Fig. 9 Solidus temperature of predicted results and the testing data

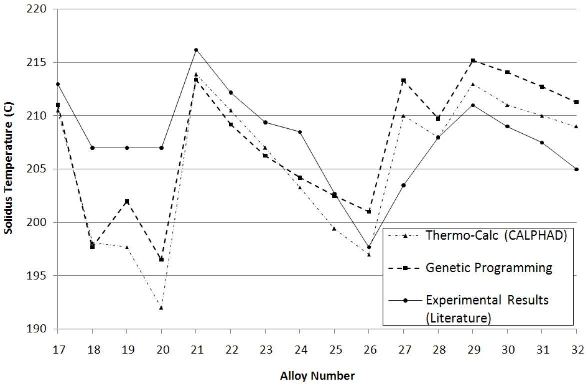

Fig. 10 Solidus temperature of predicted results and the testing data

7. Conclusion

This paper proposed the model of genetic programming to predict liquidus/solidus temperature of any Sn-Ag-Cu-Bi-In solder compositions. The prediction accuracy is very high both for liquidus training data set and solidus training data. However, the optimal search performance is also dependent on the parameter setting of the algorithm.

The best solution of liquidus prediction was obtained by the following parameters: the population size to 50; tournament size 4; maximum generation 500; probability of reproduction, crossover, and mutation 0.1, 0.7, and 0.25, respectively.

Meanwhile, the best solution for predicting solidus temperature has the following parameters setting: population size 50; tournament size 7; maximum generation 500; probability of reproduction, crossover, and mutation 0.1, 0.9, and 0.25, respectively.

The model is verified by the experimental data from literatures and the result showed very good prediction for both liquidus/solidus temperatures.

The root mean square error between experiments and genetic programming was 10.6239 in liquidus prediction and was 4.1696 in solidus prediction. Moreover, both of them had a lower RMSE when compared to the RMSE between experiments and CALPHAD approach (12.0956 in liquidus and 4.3665 in solidus).

The agreement of the predicted results and the experimental data shows that genetic programming is a powerful algorithm to predict the thermal properties of solder, melting point and solidification range. The melting point of solders could represent by the liquidus temperature and the solidification range was the different between solidus and the liquidus temperature.

With genetic programming approach, the obtained solution of this problem (mathematical expression) will help metallurgist to known the relationship between the solder alloy compositions and their properties and it will be benefit for designing new lead-free solder alloys without trial-and-error experiments.

8. Acknowledgement

This research was supported by the Technology Management Center (TMC) and the Thailand Research Fund (TRF).

9. References

[1] K. Suganuma, "Advances in Lead-free Electronics Soldering," Current Opinion in Solid State & Materials Science, 5, 2001, pp.55-64.

[2] M. Brezocnik, M. Kovacic, and M. Ficko, “Prediction of Surface Roughness with Genetic Programming,” Journal of Materials Processing Technology, 157-158, 2004, pp.28-36.

[3] M. Kovacic, P. Uratnik, M. Brezocnik, and R. Turk, “Prediction of the Bending Capability of Rolled Metal Sheet by Genetic Programming,” Journal of Materials and Manufacturing Processing, 22(5), 2007, pp.634-640.

[4] M. Li, G. Liu, and Y. Zhao, “Forecasting GDP Growth using Genetic Programming” International Conference on Natural Computation, China, 2007, pp. 393-397.

[5] J. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection,” The MIT Press, MA, 1992.

[6] N. Saunders and A.P. Miodownik, “CALPHAD: Calculation of Phase Diagrams,” Pergamon Materials Series vol.1, Elsevier Science, Oxford, 1998.