Designing Polymer Blends Using Neural Networks, Genetic Algorithms, and Markov Chains

N. K. Roy[1,2], W. D. Potter[1], D. P. Landau[2]

[1]**Department of Computer Science**
University of Georgia, Athens, GA 30602

[2]**Center for Simulational Physics**
University of Georgia, Athens, GA 30602

ABSTRACT

In this paper we present a new technique to simulate polymer blends that overcomes the shortcomings in polymer system modeling. This method has an inherent advantage in that the vast existing information on polymer systems forms a critical part in the design process. The stages in the design begin with selecting potential candidates for blending using Neural Networks. Generally the parent polymers of the blend need to have certain properties and if the blend is miscible then it will reflect the properties of the parents. Once this step is finished the entire problem is encoded into a genetic algorithm using various models as fitness functions. We select the lattice fluid model of Sanchez and Lacombe[1], which allows for a compressible lattice. After reaching a steady-state with the genetic algorithm we transform the now stochastic problem that satisfies detailed balance and the condition of ergodicity to a Markov Chain of states. This is done by first creating a transition matrix, and then using it on the incidence vector obtained from the final populations of the genetic algorithm. The resulting vector is converted back into a

population of individuals that can be searched to find the individuals with the best fitness values. A high degree of convergence not seen using the genetic algorithm alone is obtained. We check this method with known systems that are miscible and then use it to predict miscibility on several unknown systems.

I INTRODUCTION

Several techniques ranging from molecular dynamics to lattice models exist to model polymer systems but none can utilize the vast amount of data available for polymers. While these have met with some success when one considers simple linear polymers they do not give accurate results when diverse systems like branched polymers, high molecular weight polymer mixtures, block copolymers, interpenetrating networks, polymer dispersed liquid crystals, and cross-linked systems are studied. Added to this is the unreasonably large computation time required to model sufficiently large systems. The synthesis of new polymers has reached a point where the chemical synthesis of a new mer (the basic repeat unit in a polymer) is very rare. Thus the only other way to design new materials without involving expensive chemical synthesis such as in block copolymers is to obtain in most cases miscible or compatible polymer blends, or in some cases control over the degree of immiscibility as in polymer dispersed liquid crystals. However in high polymers like engineering plastics compatibility is an exception rather than a rule. This severely limits the development of new polymer systems with useful mechanical, optical, electrical and thermal properties. The use of compatibilizers, plasticizers, or anti-plasticizers (almost always a low molecular weight component) does achieve the desired result of compatibility in many cases. However the search space for such combinations is very

large and without a tool to predict the outcome, attempts to proceed with the myriad of polymers, oligomers, and resins reduces statistically to a trial and error process. This is the reason for exploring new methods to design polymer blends.

The key objective of thermodynamic modeling of polymer-polymer and polymer-solvent systems is to predict, or at the very least correlate the phase behavior in liquid, melt or solid phase. It may then be possible to predict for example whether or not a particular polymer blend will form a homogeneous (miscible) or heterogeneous structure, as when cooled from a melt, or if cast from a common solvent as the solvent evaporates. In any case the properties of the parents are important, as the blend if miscible will exhibit the properties of the parent[1,4]. *Hence in order to predict the properties of modified mers (basic repeat units) we use Neural Networks (NN) that have received extensive training*. There are a wide variety of nets[2] available and this coupled with the vast polymer database at our disposal makes for a powerful method in itself for polymer design. It may be mentioned that modifying a mer is chemically more feasible than trying to polymerize a new polymer starting with a new mer. The modification involves replacing one or more atoms on some pendant side group leaving the backbone untouched. One can thus create new polymers ranging from simple modified ones to ionomers that have excellent potential to be realized chemically in a cost effective manner[3-7].

This first step leaves us with polymers that are now candidates to be studied for blending. *In the second step, the encoding of the problem into a genetic algorithm[8] (GA) is done and then the GA is run*. We restrict the problem to binary systems. The extension to ternary

and higher order systems is straightforward. The GA can be used with several fitness functions, depending on the model and the properties to be investigated. Thus the fitness function can be a simple solubility parameter[9], or a more complicated interaction parameter[10], or even the Gibbs free energy of mixing[11].

After the GA is run and reaches a steady-state, the third step involves obtaining a transition matrix using the final population of the GA. This transition matrix is used to obtain a Markov Chain of states. From this then a transformation is made to obtain the new final population. This leads to the values of the fitness function (maxima or minima) with the extremely high convergence needed that conventional GA's cannot provide. *By studying the results (values obtained for the fitness function and its parameters) one can then predict quite accurately the polymer blend properties*. This is shown first for two known systems and then for several unknown systems.

II MODELING METHODS AND RESULTS

Neural Network Modeling:

i) Network Training and Testing Phase

A common goal of materials science is the determination of relationships between the structure (microscopic, mesoscopic and macroscopic) of a material and its properties (mechanical, thermal, magnetic, optical, electrical, environmental, and deteriorative). This information is crucial for engineering materials that provide a pre-determined (required) set of properties. One technique other than designing a whole new polymer with a given set of properties in mind is the technique of polymer blending where two or more

polymers are used to form a compatible (thermodynamically miscible) blend, such that the resultant blend has a dominance of favorable properties of the parent polymers. However in nature compatibility in high/high molecular weight polymers is an exception rather than a rule and this severely restricts the development of good engineering plastics. The use of compatibilizers, plasticizers and anti-plasticizers with techniques such as block copolymerization offer a viable alternative but these techniques still have limitations when dealing with most major engineering polymers and suffer from the drawback of involving much trial and error. The prediction of polymer properties from just the structure of the monomer is complicated. However trained neural networks given optimized input data do an excellent job of characterizing a new modified polymer that can then be easily synthesized.

While many properties are desired in an engineering plastic, probably the most important is its impact resistance. One indicator of good impact resistance is the T_α / T_γ ratio[12]. The higher this value the higher the impact resistance of the polymer. This along with the dynamic elastic modulus of the polymer can characterize very accurately its mechanical properties. T_α or glass-transition temperature[13] is the temperature at which the onset of long-range segmental mobility occurs. T_γ is a lower order relaxation temperature associated with motions in the side-chains[14]. While reinforcing a polymer and creating cross-linked or composite materials gives extremely good materials there is a sacrifice in the optical properties. One such polymer (bisphenol -A polycarbonate[15] or PC) commercially available as Lexan/Makrolon/Calibre is widely used in bullet-proof glass. Its T_α / T_γ ratio is 2.5 and dynamic elastic modulus at $20^{O}C$ is 5.02 x 10^9 dynes/cm^2.

Another class of polymers are poly-phenylene oxides with this ratio being typically 1.7 for poly(2,6-dimethyl-1,4-phenylene oxide[16]) and dynamic elastic modulus at $20^{O}C$ being 6.21×10^{9} dynes/ cm^2. In this work we restrict our studies to these two polymers, modifications of these, blends of modifications of PC with poly(methyl methacrylate)[17] (PMMA) (PC is known to be miscible with PMMA for PC greater than 50% concentration ranges[18]), and blends of modifications of PPO with poly(styrene)[19] (PS) (PPO is miscible with PS over the entire composition range[20]). PPO/PS blends are commercially available as Prevex/Noryl.

The Neural Networks used ranged from the standard type of Backpropagation[6] network in which every layer is connected or linked to the immediately previous layer with the option of using a three, four, or five layer network (with one, two, or three hidden layers respectively), recurrent networks with dampened feedback (Jordan-Elman nets)[21], multiple hidden slabs with different activation functions (Ward nets)[22], nets with each layer connected to every previous layer (jump connection nets)[23], unsupervised Kohonen nets[24], Probalisitc nets[25], the General Regression net[26], and Polynomial nets[27]. The activation functions[6] ranged from the standard logistic and symmetric logistic, to linear, tanh, sine, Gaussian and Gaussian complement. It has been shown[2,6] that the simple Backpropagation network with at most 2 hidden layers can solve any non-linear problem provided there are a sufficient number of hidden nodes.

The various polymers (440 in all) in the polymer database which includes a wide variety of polymers are given in Table 1. This includes all types of polymers. The number in

parentheses indicates the total number of different polymers from that class. The database fields include (all kinds of bonds are counted.): number of C-C (Carbon-Carbon) single bonds; number of C-H (Carbon-Hydrogen) single bonds; number of C-C (Carbon-Carbon) double bonds; number of C-C (Carbon-Carbon) triple bonds; number of O (Oxygen) bonds; number of N (Nitrogen) bonds; number of P (Phosphorous) bonds; number of S (Sulfur) bonds; number of cyclic rings; number of halide bonds; number of Si (Silicon) bonds; aspect ratio; 3-D Wiener Number; molecular weight (number and weight average); Dynamic Elastic Modulus at room temperature (20^{O}C); Glass Transition Temperature (T_{α}) and lower secondary order transition temperatures T_{β} , T_{γ} , and T_{δ} . The total number of inputs for training the different nets were:

I1 = Number of C-C (Carbon-Carbon) single bonds;
I2 = Number of C-H (Carbon-Hydrogen) single bonds;
I3 = Number of C-C (Carbon-Carbon) double bonds;
I4 = Number of C-C (Carbon-Carbon) triple bonds;
I5 = Number of O (Oxygen) bonds;
I6 = Number of N (Nitrogen) bonds;
I7 = Number of P (Phosphorous) bonds;
I8 = Number of S (Sulfur) bonds;
I9 = Number of cyclic rings;
I10 = Number of halide bonds;
I11 = Number of Si (Silicon) bonds;
I12 = Aspect Ratio;
I13 = 3-D Wiener Number;
I14 = Molecular Weight (Weight Average).

While all the above are self-explanatory, aspect ratio and 3-D Wiener Number need special mention. The aspect ratio is a measure of asymmetry in a monomer (polymer repeat unit) and is the ratio of the length of the long axis to the short axis of the monomer. Monomers are three-dimensional objects and hence in addition to their topological and combinatorial contents their 3-D character is of profound importance. The 3-D Wiener number is based on the 3-D (geometric, topographic) distance matrix, whose elements

represent the shortest Cartesian distance between two i-j pairs. The matrix is real and symmetric. Details in the way the extraction of this number from the matrix is done are given in reference [28]. The exact outputs provided were O1= T_α / T_γ, and O2=Dynamic Elastic Modulus. The question about reducing the inputs was not considered as all the inputs were extremely important and could not be discarded if theoretical accuracy in the modeling was desired.

In all, the original pattern file had 440 patterns. 80% were extracted at random to form the training set, which now had 352 patterns. 20% were then in the testing set (88 patterns). It was found that in this case the multi-layer Backpropagation network gave the best results as compared to the other types of networks mentioned above. Hence we present results only for this type of network. Figure 1 gives the summary of results for a single hidden layer network. The number of calibration events was 50, and training was saved based on the best test set. The training ended when the number of events since the minimum average error exceeded 500000. Weight update selection was momentum. Pattern Selection was random during training. On analyzing the data and noting that r^2, the mean squared error, and the correlation coefficient were best for 17 hidden nodes and as a very high number of hidden nodes did not appear to improve these values further, the number of hidden nodes was fixed at 17. In this region the 'Percent above 30%' (of the actual values) also showed a minima while the correlation coefficient exhibited a maxima (Figure 1). As will be shown, the addition of a second hidden layer did not improve the results. Notice that the N=17 case gave better results than the N=100 case for the 'Percent within 5%' and the 'Percent over 30%' values, although the N=100 case gave marginally

better R^2, r^2, and correlation coefficient values. The N=17 case had the lowest mean squared error and mean absolute error. Given such narrow differences made making choices difficult, but finally considering the fact that higher nodes (N=100 case) may cause redundancy in many weights, and that the N=17 case did have the lowest mean square error, mean absolute error, and better 'Percent within 5%' and 'Percent over 30%' values as compared to the N=100 case, it was selected. It also had the second highest value for the correlation coefficient. The graphs in Figure 1 tended to flatten out after N=15, suggesting some statistical fluctuations for N > 15, and indicating that the optimum number of nodes for accurately defining the given problem was reached.

The bottom-up technique for selecting the number of hidden nodes was used in favor of the top-down method. Genetic algorithms or other combinatorial optimization methods could also be effectively used, but in terms of simplicity this was the best approach. Figure 2 shows the results obtained when varying the initial weights from 0.3. The number of hidden nodes was now fixed at 17. All other parameters including the learning rate and momentum were the same as in the case above. As compared to the case of 0.3 for 17 nodes in Figure 1 all the values for correlation coefficient, r^2, and mean squared error are poorer. Figure 3 gives the results of varying the momentum with learning rate fixed. All other parameters were the same as in the first case with the number of hidden nodes fixed at 17. A momentum of 0.4 gave only a marginally poorer value of correlation coefficient, r^2, and mean squared error as compared to the selected value of 0.5. Very low or higher values gave poor results. Figure 4 gives the results of varying learning rate keeping momentum fixed at 0.5. Number of hidden nodes was 17, and all other parameters were

the same as in the first case. Increasing the learning rate caused further deterioration in the values of correlation coefficient, r^2, and mean squared error, from gradual to more rapid as the learning rate increased.

Finally adding a second hidden layer to the network did not show any improvements as can be seen from Table 2. All parameters are the same as in the first case. The number of hidden nodes in the first hidden layer is 17. While increasing the number of hidden nodes in the second layer did improve the results they were in all cases poorer than the final selected model (R^2, r^2, correlation coefficient r, and mean squared error values.). We also tried varying the number of hidden nodes in the second layer while setting the number of hidden nodes in the first layer to 5, 10, 20, 50, and 100. In all cases the results were worse than the case discussed here. Figure 2(a) and Figure 2(d) show the reason an initial weight of 0.3 was chosen in the final model. This value also gave the smallest mean squared error (Figure 2(c)), and highest 'Percent < 5%' values Figure 2(b). A momentum of 0.5 (Figure 3(a)) gave a maxima, and the mean squared error was minimum (Figure 3(c)). For this value as seen in Figure 3(d) the graph flattened out. From Figure 3(b) we see that 'Percent < 5%' gave a maximum with a corresponding minimum for 'Percent > 30%'. Thus this value appeared to be an optimal number and was selected in the final model. Figure 4 shows that a choice of learning rate of 0.05 leads to the deterioration of the results with an increase in learning rate.

The final selected model details are as follows: standard backpropagation single hidden layer neural network, 17 hidden nodes, 14 input nodes, 2 output nodes, logistic activation

function, logistic scaling function, weight updates is momentum, pattern selection is random, initial weight is 0.3, learning rate is 0.05, and momentum is 0.5. The min/max values of each of the input variables were set by scanning the input before being applied to the network. File extraction was random. Figure 5 gives a scatter plot of the actual versus the predicted output when the trained network was applied to the production file with 88 patterns randomly selected from the 440 patterns in the database where the exact values are known and can be compared with the predicted values from the final model. Note that this set differed from the 88 pattern testing set.

ii) Application Of Final Trained Network To Predict Properties:

The trained model was now presented with inputs I1-I14 calculated specifically from 24 different modified bisphenol - A polycarbonates and 19 different modified poly(2,6-dimethyl-1,4-phenylene oxide) monomers. The range of each of the inputs was checked to be between the maximum and minimum values of the respective inputs in the 440 pattern set of the polymer database before these inputs were applied to the final selected network. Five candidates for modified polycarbonates were found and two for modified poly(phenylene oxides) from these. Table 3 gives the results. Figure 6 gives their structures. The feasibility of this method depends on correct identification of structure-property relationships in the monomers that form the input and the chemical viability of the synthesis such that the targeted new monomer (polymer) has dimensional stability, ease of production, and cost effectiveness. These modified polymers all have higher values for the T_α / T_γ ratio, and the Dynamic Elastic Modulus as compared to the parent polymers. These are now used for simulations using the GA-Markov Chain technique where we consider blends of the modified PC's with PMMA, and modified PPO's with PS.

B Genetic Algorithm - Markov Chain Modeling:

i) Encoding and running the Genetic Algorithm for Polymer Blend Modeling

To encode the problem of polymer blend design, several factors have to be considered. The most important of these is the fitness function, followed by the cross-over[29] scheme, the mutation and cross-over probabilities[29], the size and cardinality of the individual strings in the population, and the size and initial fitness values of the starting population. Due to convergence, the final population represents one with many individuals with high fitness values or in other words one with many possible solutions. Further for our technique to be effective the genetic algorithm must reach a steady state (in preparation for the Markoc Chain phase). By steady state it is meant that the value of the fitness function and its parameters fluctuate within a statistical limit presenting several optimal solutions in the process of maximizing or minimizing the fitness function. While this will occur if the simulation is run for a large amount of time one must consider that in reality the simulation must result in the average fitness values of the population in the given generation to reach a constant asymptotic value or be within a certain range with only slight fluctuations in the highest fitness value or in the range of fitness values of an individual in the population of a given generation in time that is computationally feasible. Further care must be taken to ensure that the system is not trapped in a local maximum (or minimum). The use of simulated annealing[31] will only serve to lengthen and complicate the process. Thus the solution is to use trial and error in which several simulations are done starting with different population sizes and initial average fitness values, selecting different values of the cross-over and mutation probabilities, and then running the GA until convergence (average fitness values become constant or within a certain range of

preset fluctuation values and the highest fitness values of the individual in successive generations reaches a constant or is in a certain range). By doing this on a known system (in this case the PC/PMMA blend and the PPO/PS blend) good parameters can be selected.

The most important consideration in predicting polymer blend miscibility is the Gibbs free energy of mixing (g)[1,4,32]. One of the best thermodynamic models known is the lattice fluid theory of Sanchez and Lacombe[1,4,32-34] for binary polymer mixtures. Extending this model to higher order systems is trivial[35]. Unlike the famous Flory-Higgins[36] theory, this theory allows the lattice to be compressible and have vacant sites. Thus this model is able to describe volume changes on mixing. In this approach one defines g as:

$$g = g_c + g_{nc} \qquad \ldots(1)$$

where g_c and g_{nc} are the combinatorial and non-combinatorial parts of the free energies per mer of mixture respectively.

$$g_c = kT\left[\left(\frac{\phi_1}{r_1}\right)\ln(\phi_1) + \left(\frac{\phi_2}{r_2}\right)\ln(\phi_2)\right] \qquad \ldots(2)$$

and

$$g_{nc} = -\tilde{\rho}\varepsilon^* + \tilde{\rho}\tilde{v}v^* + kT\tilde{v}\left[(1-\tilde{\rho})\ln(1-\tilde{\rho}) + \left(\frac{\tilde{\rho}}{r}\right)\ln(\tilde{\rho})\right] \qquad \ldots(3)$$

where

$$\varepsilon^* = \phi_1^2\varepsilon_{11}^* + 2\phi_1\phi_2\varepsilon_{12}^* + \phi_2^2\varepsilon_{22}^* \qquad \ldots(4)$$

$$v^* = \phi_1^2 v_{11}^* + 2\phi_1\phi_2 v_{12}^* + \phi_2^2 v_{22}^* \qquad \ldots(5)$$

$$\frac{1}{r} = \frac{\phi_1}{r_1} + \frac{\phi_2}{r_2} \qquad \ldots(6)$$

and $\tilde{\rho}$, and $\tilde{v}$ are the reduced density[37,38] and volume[37,38] of the mixture respectively at 20^{O}C. ϵ_{ij}^{*} and v_{ij}^{*} are temperature dependent energetic parameters that characterize the attractive and repulsive parts of the mer-mer interaction potentials[39,40]. Note for binary systems if i $\neq$ j, $\varepsilon_{ij}^{*} = \varepsilon_{ji}^{*}$ and $v_{ij}^{*} = v_{ji}^{*}$. r_1 and r_2 are the chain lengths of polymer 1 and 2 respectively, and ϕ_i is the volume fraction of component i in the mixture. Further two dimensionless interaction energy parameters[32-34,39,40] can now be defined as:

$$\zeta = \varepsilon_{12}^{*} / (\varepsilon_{11}^{*} \varepsilon_{22}^{*})^{1/2} \qquad \ldots(7a)$$

and

$$\xi = v_{12}^{*} / (v_{11}^{*} v_{22}^{*})^{1/2} \qquad \ldots(7b)$$

The dimensional interaction energy parameters above give a normalized view of the temperature dependent attractive and repulsive parts of the mer-mer interaction parameters. They include the respective self-attractive and self-repulsive parts, and provide a way of comparing different polymers and their binary interactions in a self consistent manner. The values of these interaction energy parameters for PC, PS, PPO, PMMA, and the five modified PC's and two modified PPO's and their blends for different weight ratios all at room temperature (20^{O}C) are given in Table 4. These are calculated exactly using the structure of the respective mers by the UNIFAC technique[41,42] that involves consideration of the functional groups rather than molecules, the molecular weight of the polymers, and in the case of blends, the weight ratio of the parent polymers. The formalism has been extended to allow for mixtures of both small molecules and polymers to be treated simultaneously. For miscible blends the necessary but not sufficient condition for miscibility is that $g < 0$[4]. With this constraint once the GA has reached a steady state, the value of the dimensionless interaction energy parameters can be found

and compared with the calculated ones. Note we present examples for blend ratios that did reach a steady state and that did give final generations in which all the individuals had $g < 0$. For various weight ratios we ran the GA for a maximum of 2×10^6 generations. If by this time the GA did not reach a steady state we banded the system as immiscible and rejected it for further study. Thus the equation for g (which has to be minimized) forms the fitness function with the constraint that for the steady state g must become negative for all the individuals in the given population in the given generation and for subsequent generations. A somewhat unusual GA convergence scheme.

To find useful values of mutation (p_m) and cross-over probabilities (p_c) several runs were done for different values of p_c and p_m with the population size fixed at 1000 individuals. The simulation proceeded for 2×10^6 generations by which time every individual had g negative, which we define as the condition for a steady state. After this the next 1000 generations were saved (every individual in these generations had $g < 0$) for use in forming the transition matrix described below. Table 5 gives the results. In order to be sure that running the GA for more time would give an additional improvement we evaluated the standard deviation of g for the last 1000 generations and computed the variance. We found the variance was small if the GA did reach a steady state. The variance values are also listed in Table 5. From the variance values we see that there is a very small fluctuation in the g values. At this point one is likely to question the need for the Markov Chain method. However, g is only a necessary but not sufficient condition for miscibility and a large class of immiscible blends have g negative. What is important however are the values of the dimensional interaction energy parameters. We found this

from the best fit individual in each of the 1000 generations sampled after a steady state was reached and before the Markov Chain method was applied. The error bars in this case were over four orders of magnitude higher than that shown in Table 5. This was just not acceptable. Thus the GA method alone gave large fluctuations in the values of the interaction parameters although the g values appeared to converge (reach a steady state). A further screening (convergence) in the individuals was required, so that the error bars in the values of the interaction parameters would be within acceptable limits. We could not achieve this using the GA method alone after extensive simulations.

For prediction of miscibility the interaction parameters are the most important and need to be predicted with a very high degree of accuracy. If we could not reach a steady state for the blend system for a given weight ratio then it can be safely assumed to be immiscible and we do not present results for these cases. We found that a value of $p_c = 0.3$ and $p_m = 0.05$ gave the fastest convergence to negative g and thus were used. The simulations were done at room temperature with mono-disperse systems having chain lengths of both the component polymers (r_1 and r_2) fixed at 50000 repeat units. The systems studied were known systems PC/PMMA 70/30 and PS/PPO 50/50. The corresponding values of $\tilde{\rho}$ (reduced density), and $\tilde{v}$ (reduced volume), for these two systems at room temperature for the given molecular weight and weight ratio that were calculated using equations 8 and 9 below, were taken from references [43,44], and are not reproduced.

The binary string individuals used 25 bits for each variable. The first bit was the sign bit, and the decimal point was assumed to be after the fifth bit. Since temperature was fixed at

20^{O}C and k (Boltzmann contant) was a constant, the known variables were: ϕ_1, ϕ_2, r_1, r_2, $\tilde{\rho}$ and $\tilde{v}$ that were directly used in finding g. However there were six unknown variables which were all positive. The total length of an individual was thus 150 bits.

Two-point cross-over[29] between two randomly selected individuals in a population was used and the bits were exchanged only between similar variables in the individual pairs selected for reproduction and cross-over. Selection of individuals from the previous generation to form a new generation involved keeping 80% of the individuals with the best fitness values from the previous population and padding the remaining 20% with individuals with the highest fitness values from this previous generation. Since the population size was fixed at 1000, the number of individual pairs for cross-over was randomly selected and fixed at 500. Similarly the number of attempts at mutation in each generation was fixed at 1000 where each individual was chosen at random for an attempt at mutation. For cross-over to occur a random number was called (for each paired individual in the population) between 0 and 1, and compared with the cross-over probability. If it was less than the cross-over probability, cross-over was done as follows. First two points (starting positions for the cross-over) were randomly selected in each of the corresponding 25 bit variable positions in the bit string. These represented the same variable in the two individuals. Then a random number from 1 to 25 determined the length (ending point of the cross-over). Periodic boundary conditions were used for each of the six 25 bit substrings in the 150 bit long individual. After this the bits were exchanged and the next pair of individuals was considered. Hence each cross-over operation involved 6 pairs of points on each of the two selected individuals.

For mutation the individual was chosen at random. Then for each variable in the bit string a position was chosen at random. Then a random number was called between 0 and 1. If this number was below the mutation probability the value of the bit at that position was complemented. This was done six times for each individual, as there were six variables. Thus each mutation operation involved 6 points, each randomly selected in the corresponding 25 bit segment of the variable. After selection, cross-over, and mutation, the new generation was created (reproduction) from the previous one as described above.

For the values of p_c, and p_m indicated above the final population was searched and the best values for ζ and ξ found for PC/PMMA 70/30 blend were 0.997 and 0.00738 respectively, and for PS/PPO 50/50 blend were 0.942 and 0.00794 respectively. These values improved considerably after using the Markov chain modeling technique as will be shown below. For modeling blends of PMMA with modified PC and blends of PS with modified PPO, we use the fact that for high polymers in bulk (solid state at room temperature ~ 20^{O}C) the values of $\tilde{\rho}$ and $\tilde{v}$ for most engineering plastics are in a narrow range hence the values for the corresponding known pure component of the given blend ratio are used according to the equations:

$$\tilde{\rho} = a_1 w_1 \tilde{\rho}_1 + a_2 w_2 \tilde{\rho}_2 \qquad \ldots(8)$$

and

$$\tilde{v} = a_1 w_1 \tilde{v}_1 + a_2 w_2 \tilde{v}_2 \qquad \ldots(9)$$

where for monodisperse systems $a_1 = a_2 = 1$, w_1 and w_2 are the weight ratios of component 1 and 2 in the blend respectively, and $\tilde{\rho}_i$ and $\tilde{v}_i$, are values for the pure component i in the

blend. ϵ_{ij}^{*} and v_{ij}^{*} on the other hand are very sensitive to monomer configuration and type of functional groups[45], molecular weight and blend ratios. For these systems the GA parameters were the same as for the case of the two blends described above. We present only the results obtained after the Markov chain method is used when the blends with the modified PC's and PPO's were investigated and on running the GA to a steady-state.

ii) Markov Chain Modeling Technique

Mathematical models have been used as analytical tools in the investigation of genetic algorithms. An early work is that of Goldberg's on the minimal deceptive problem[46] that resulted in the familiar type-I and type-II classifications. Vose and Liepins simplified and extended this further[47]. Holland questioned the relationships between GA's and infinite population models[48]. Real GA's are based on finite populations and are stochastic, not deterministic. The relationship between the trajectory of expectations (which is the path infinite population models followed), and the evolution of finite populations in real GA's was studied by Nix and Vose[49]. An exact model for real GA's was obtained in the form of a Markov chain, and it was shown that the trajectory followed by the finite population model was related to the evolutionary path predicted by the infinite population model. Roughly speaking, populations correspond to points on a smooth "GA-surface", and the progression from one generation to the next forms a path leading to a local minimum of the surface (downhill ~ increasing fitness). Therefore, at least in the infinite population case, genetic algorithms should not be thought of as global optimizers. Populations move quickly downhill to be trapped in some local basin. Vose showed[50] that for large populations the evolutionary path of a real GA follows very closely with large probability, and for a long period of time that path predicted by the infinite population model. Thus

the local geometry of the GA-surface indicates the transient behavior of a real GA, i.e., that behavior which depends on the initial population[51]. When mutation is nonzero (which we assume throughout) a real GA forms an ergodic Markov chain, visiting every other state infinitely often[52]. Hence in some period of time, a GA will escape every local minimum of the surface giving a probability distribution that converges to the steady state distribution of the Markov chain. Vose showed[50] that the steady state distribution concentrates probability near fixed points of the infinite population model, i.e., near the local minima of the GA-surface. So a real GA will escape one local minimum only to be trapped temporarily in another. These local minima are however not equally likely. A real GA will with large probability be asymptotically at that local minimum having the largest basin of attraction. As population size grows, the probability of a GA being asymptotically anywhere else converges to zero. Therefore, the GA-surface provides a geometric interpretation of genetic search that explains population trajectories. Short-term behavior is determined by that minimum having the largest basin, however the non-linear mathematics of GA's are complex.

Consider the case of a simple GA which finds an individual, i, that maximizes or minimizes the fitness value, $f(i)$, for the case of the fitness function in the GA polymer design problem. A population consists of M individuals

$$a_{i1}a_{i2}a_{i3}...a_{iL} \in A^L \qquad \ldots(10)$$

of length L, where

$$A = \{0,1,\ldots,\alpha\text{-}1\}\ (\alpha \geq 2). \qquad \ldots(11)$$

In this problem $\alpha = 2$. Each individual i is represented as an unsigned integer

$$\sum_{j=1}^{L} a_{ij}\alpha^{L-j}. \qquad \ldots(12)$$

Each population, k, is indicated by the incidence vector

$$(Z(0,k), Z(1,k), \ldots Z(\alpha^L - 1, k)), \qquad \ldots(13)$$

where $Z(i,k)$ denoted the occurrences of the individual labeled as $i = 0,1,...,\alpha^L - 1$ in the population k. In this case L=150. Since L is large the entire population that can be created cannot be used, but a sampling of the population generated using the standard GA with high fitness values is used to generate the transition matrix. Iterating the generation changes n times after running the GA until high population values are reached gives a Markov chain which generates a population labeled as k = 1,2,…,N after n generation changes. From this Markov chain the transition matrix is obtained. The important problem here in order to get answers comparable or even better than when using a standard GA procedure is choosing the size and point when sampling of this initial population is done. The initial population and subsequently the transition matrix are sensitive to the choice of mutation probabilities. Borrowing from Suzuki[55] for the mathematical representation, "Vose, and Nix and Vose have showed[49,50] that the stochastic transition through the genetic operations (meaning selection, cross-over, mutation) can be fully described by the transition matrix

$$Q = (Q_{k,v}) \qquad \ldots(14)$$

of size NxN for one generation, where $Q_{k,v}$ is the conditional probability that population v is generated from population k. When the number of generated individuals equals M (M: even number) the following equation is obtained:

$$Q_{k,v} = M! \prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!} r(j,k)^{Z(j,v)}. \qquad \ldots(15)$$

because Z(j, k) is generated according to the multinomial distribution based on r(j, k), j = 0, 1, … , 2^L - 1, where r(j, k) is the probability that individual j occurs in population k". The approach in obtaining r(j, k) takes many forms, details are given in reference [47]. However, Vose, and Nix and Vose showed[49,50] in terms of the mutation probability μ ($0.5 < \mu < 1$) that:

$$\mu^{LM} M! \prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!} \quad \leq \quad Q_{k,v} \quad \leq \quad (1-\mu)^{LM} M! \prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!} \qquad \ldots(16)$$

where it is assumed for simplicity that the cross-over probability is set to one. The inequality signs in equation 16 are reversed when $0 < \mu < 0.5$. For $\mu = 0.5$ the equality sign holds, as this is a crossover point. For a cross-over probability of 1 this point (0.5) is symmetrical and the two regions below and above 0.5 are identical. One can use any region self-consistently, however for a cross-over probability of less than 1 the expression for the two bounds is much more complicated as the symmetry is broken and the easy reduction to inequality 16 in not possible[49]. Hence a reasonable approximation is to take the value of $Q_{k,v}$ as the average of the two upper and lower bounds. To compute the two bounds which essentially involves multiplications and divisions we take each term in the product $\prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!}$ and multiply it by one term of the constant factor and one term of the factorial (M!) until all the terms in $\prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!}$ are included. Remember we had to truncate the upper bound $2^L - 1$ as this is too large a number to be computationally

feasible. This prevents underflow errors that will result if the constant factor outside the product $\prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!}$ is calculated first. This has to be repeated for the calculation of each of the bounds in inequality 16 separately. We cannot directly evaluate the term $\prod_{j=0}^{2^L-1} \frac{1}{Z(j,v)!}$ and then multiply it with the constant factor μ^{LM} or $(1-\mu)^{\mathrm{LM}}$ and M!, as the constant factors μ^{LM} and $(1-\mu)^{\mathrm{LM}}$ if computed separately give rise to underflow errors as the numbers are too small. Using the average of the two bounds also has the advantage over other methods that are computationally far more intensive. It has been shown for a proper selection of an initial population (high fitness values) that this approximation works very well[54, 55]. Once the transition matrix is obtained, from an (again following Suzuki[55]) "initial distribution of N possible populations:

$$q^{(0)} = (q_1^{(0)}, q_2^{(0)}, ..., q_N^{(0)}), \quad \ldots(17)$$

the distribution of each population after a generation change is simply:

$$q^{(1)} = (q_1^{(1)}, q_2^{(1)}, ..., q_N^{(1)}) = q^{(0)}Q. \quad \ldots(18)$$

After n generation changes we get:

$$q^{(n)} = (q_1^{(n)}, q_2^{(n)}, ..., q_N^{(n)}) = q^{(0)}Q^n \quad \ldots(19)$$

in terms of the nth power

$$Q^n = (Q_{k,v}^{(n)}) \quad \ldots(20)$$

of the transition matrix Q". Hence all that needs to be done is to multiply Q some n times and apply it to the initial population distribution.

The resultant vector can now be searched for the individual with the highest fitness, which is an easy task. Since L=150, and $\alpha = 2$, $\alpha^L - 1 \sim 1.427 \times 10^{45}$ which is an extremely large number. This was truncated to 1000. The number of individuals in each population was fixed at 1000. After g became negative for all individuals in the generation the next 1000 generations were saved in order to determine the transition matrix Q as described above. The exponent n was set also to 1000. From the final vector obtained ($q^{(1000)}$) the final 1000 individual generation is reconstructed and searched for the individual with the best fitness value. Table 5 gives these results. Error bars are from five separate simulations starting with a new random generation in the GA modeling phase. These can now be compared with the exact values for these systems in Table 4. Further since g is negative for all these cases we have also predicted that these systems should be miscible for the given compositions. It must be pointed out that if the case of immiscibility was obtained, one way to design (at least theoretically) a miscible blend would be to extend this to a three component system, with the third component such as a low-molecular weight resin, organic molecule, or an oligomer. These species can act like compatibilisers[56] in effect tuning the ϵ_{ij}^* and v_{ij}^* to more favorable values resulting in miscibility. Simply by studying the values of ϵ_{ij}^* and v_{ij}^* for the immiscible system that is predicted by this method one can select the right compatibiliser to obtain favorable values for ϵ_{ij}^* and v_{ij}^*.

III CONCLUSIONS

The field of polymer blend design is an extremely diverse and complex area with equally diverse and complex problems. So far no single design technique is known to be the best.

However, as seen in this work there is much potential in using both Neural Networks, and GA's. As we have shown we use the neural network technique to select potential modified polymers that have better specific properties than the parents. However using the neural network method to predict miscibility on a thermodynamic scale is still not feasible. This is due to the large number of miscible and immiscible systems whose data will have to be included in the training. Further, in high molecular weight systems the sensitivity of the mer-mer interaction potentials cannot be stressed enough. This together with the large number of inputs and data needed to train a suitable network to predict miscibility reliably using neural networks is a computationally difficult task. Hence we are limited to using the neural network to accurately predict specific polymer properties rather than the complex issue of polymer miscibility.

We found that the GA method alone did not give good results. Thus once a steady state was reached and checked to be the case a transformation to a Markov Chain of states was possible as the process was stochastic and detailed balance was preserved. While the fluctuations in g are small after the steady state is reached, the values of the dimensional interaction parameters extracted from the best-fit individuals are so large (over four orders of magnitude higher than the ones gotten after using the Markov Chain method) that this is not acceptable in predicting miscibility. These large fluctuations, when compared with exact values for the known systems studied, indicate a lack of convergence using the GA method. Thus while the evolution of subsequent generations after a steady state is reached can now be considered stochastic and given that we can show that the condition of detailed balance is met there is no way to impose conditions (constraints) on either the g

values in the subsequent generations (as g value fluctuations are small) or conditions (constraints) on the energy parameter variables from which the g value is calculated (as fluctuations are very large) except for the exact cases where we know the answers and can impose such constraints to get the very high fit individuals. But this is not acceptable and one must be able to get convergence without excessively constraining the system. Thus knowing that a Markov Chain of states can describe the evolution process after steady state is reached we use this method. And we do get the convergence in the energetic interaction parameters that comprise the fitness function that the GA method alone did not produce.

The GA allows great flexibility in the choice of the fitness function, and several approaches that we have not tried yet still exist. These include using the solubility parameter approach for simulating the solvent blending technique[57], or using the values of the isothermal compressibility, thermal expansion coefficient and thermal pressure coefficient to simulate polymer melts (melt mixing[58]). Thus while the fundamental thermodynamics of mixing are invariant, the area of modeling will continue to evolve. As we have shown in this paper there will always be a strong theoretical background where exact results for known systems will be available. With these data in hand and with the improvement of computation speed the modeling methods can be tested extensively and refined and highly accurate predictions can be made.

Acknowledgement: Our appreciation goes to the reviewers and to Professor Moonis Ali for all their helpful comments and suggestions for improving this paper.

REFERENCES

[1] I. C. Sanchez and R.H.Lacombe, "Elementary Equation of State for Polymer Liquids", ***J. Polym. Sci. Polym. Lett. Ed.***, 15, 71, (1977).

[2] B. G. Sumpter, C. Getino, and D. W. Noid, "Theory and Applications of Neural Computing in Chemical Science", ***Annu.Rev.Phys.Chem.***, 45, 439-81, (1994).

[3] R. L. Jalbert and J. P. Smejkal, in "Modern Plastics Encyclopedia 1976-1977", McGraw-Hill, New York, 1976.

[4] D. R. Paul and S. Newman, Eds., "Polymer Blends", Academic Press, New York, 1978.

[5] H. V. Boenig, "Structure and Properties of Polymers", John Wiley & Sons, New York, 1998.

[6] J.Bicerano, "Prediction of Polymer Properties", Marcel Dekker, New York, 1998.

[7] J.Bicerano, Eds., "Computational Modeling of Polymers", Marcel Dekker, New York, 1998

[8] J. Reed, R. Toombs, and N. A. Barricelli, "Simulation of Biological Evolution and Machine Learning. 1. Selection of Self –Reproducing Numeric Patterns by Data Processing Machines, Effects of Hereditary Control, Mutation Type and Crossing", ***Journal of Theoretical Biology***, 17, 319-342, (1967).

[9] C. Hansen, and A. Beerbower, "Solubility Parameters", ***Kirk-Othmer Encycl. Chem. Technol.***, 2nd Ed., Suppl. p 889, (1971).

[10] B. E. Eichinger and P. J. Flory, "Thermodynamics of Polymer Solutions. 1. Natural Rubber and Benzene", ***Trans. Faraday Soc.***, 64, 2035, (1968).

[11] I. C. Sanchez and R. H. Lacombe, "Elementary Molecular Theory of Classical Fluids - Pure Fluids", ***J. Phys. Chem.***, 80, 2352, (1976).

[12] I. M. Ward, "Mechanical Properties of Solid Polymers", Wiley-Interscience, New York, 1989.

[13] I. E. Nielsen, "Mechanical Properties of Polymers", Reinhold, New York, 1991.

[14] T. Murayam, "Dynamic Mechanical Analysis of Polymeric Materials", Elsevier, 1978.

[15] J. Brandrup, E. H. Immergut, Eds., "Polymer Handbook", John Wiley & Sons, 1989.

[16] R. Juran, Eds., "Modern Plastics Encyclopedia", McGraw Hill, New York, 1989.

[17] Z. G. Gardlung, in "Polymer Blends and Composites in Multiphase Systems", C. D. Han., Eds., ***Advances in Chemistry Series***, 206, ACS, Washington, D. C., 1984.

[18] Z. G. Gardlung, "Thermal and Dynamic Mechanical Analysis of Polycarbonate / Poly(methyl methacrylate) Blends", ***Polym. Prepr. Am. Chem. Soc. Div. Polym. Chem.***, 23, 258, (1982).

[19] R. H. Boundy and R. F. Bayer, Eds., in "Styrene, Its Polymer, Copolymer and Derivatives", Reinhold, New York, 1972.

[20] E. P. Cizek, U. S. Patent No. 3,383,435 (May 14, 1968), assigned to General Electric Co.

[21] J. Dayhoff, "Neural Network Architechtures", Van Nostrand Reinhold, New York, 1990.

[22] R. Hecht-Nielsen, "Neurocomputing", Addison-Wesley, Reading, MA, 1990.

[23] T. Khanna, "Foundations of Neural Networks", Addison-Wesley, Reading, MA, 1990.

[24] T. Kohonen, "Self Organization and Associative Memory", Springer-Verlag, Berlin, 1988.

[25] D. Specht and P. Shapiro, "Generalization Accuracy of Probabilistic Neural Networks Compared with Backpropagation Networks", ***Proceedings of the International Joint Conference on Neural Networks***, 1, 887-892, July 8-12, (1991).

[26] D. Specht, "A General Regression Neural Network", ***IEEE Trans. On Neural Networks***, 2, 6, 568-576, Nov., (1991).

[27] S. Haykin, "Neural Networks – A Comprehensive Foundation", Macmillan College Publishing Company, New York, 1994.

[28] N. Trinajstic, "Chemical Graph Theory", CRC Press, London, 1992.

[29] J. H. Holland, "Genetic Algorithms and Classifier Systems: Foundations and Future Directions", ***Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms***, 82-89, (1987).

[30] H. J. Antonisse and K. S. Keller, "Genetic Operators for High-Level Knowledge Representations", ***Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms***, 69-76, (1987).

[31] L. Davis and M. Steenstrup, Genetic Algorithms and Simulated Annealing: An overview., in L. Davis, Eds., "Genetic Algorithms and Simulated Annealing", p 1-11, London Pitmann, 1987.

[32] P. Hamada, T. Shiomi, K. Fujisawa, and A. Nakajima, "Statistical Thermodynamics of Polymer Solutions Based on Free Volume Theory", ***Macromolecules***, 13, 729, (1980).

[33] I. C. Sanchez and R. H. Lacombe, "Statistical Thermodynamics of Polymer Solutions", ***Macromolecules***, 11, 1145, (1978).

[34] I. C. Sanchez, "Statistical Thermodynamics of Bulk and Surface - Properties of Polymer Mixtures", ***J. Macromol. Sci. Phys.***, B17, 565, (1980).

[35] I. C. Sanchez, in "Polymer Compatibility and Incompatibility, Principles and Practice", Jolck, Eds., ***MMI Symp. Ser.***, Vol 3, 59-76, Copper Station, N. V., Harwood, 1982.

[36] P. J. Flory, "Thermodynamics of Polymer Solutions", ***Discuss. Faraday. Soc.***, 49, 7, (1970).

[37] T. Clark, "A Handbook of Computational Chemistry: Practical Guide to Chemical Structure and Energy Calculations", New York, Wiley, (1996).

[38] P. J. Flory, "Statistical Mechanics of Chain Molecules", Wiley-Interscience, New York, 1979.

[39] S. I. Sandler, "Models for Thermodynamic and Phase Equilibria Calculations", Marcel Dekker, New York, 1993.

[40] M. A. Johnson and G. M. Maggiora, "Concepts and Applications of Molecular Similarity", John Wiley & Sons, New York, 1992.

[41] A. Fredenslund, J. Gmehling, and P. Rasmussen, "Vapour-Liquid Equilibria Using UNIFAC", Elsevier, Amsterdam, 1977.

[42] G. R. Strobl, "The Physics of Polymers", Springer-Verlag, Berlin, 1996.

[43] O. Olabisi and R. Simha, "Pressure-Volume-Temperature Studies of Amorphous and Crystallizable Polymers. I. Experimental", ***Macromolecules***, 8, 206, (1975).

[44] H. Hocker, G. J. Blake, and P. J. Flory, "Equation-of-State Parameters for Polystyrene", ***Trans. Faraday Soc.***, 67, 2251, (1971).

[45] R. H. Lacombe and I. C. Sanchez, "Statistical Thermodynamics of Fluid Mixtures", ***J. Phys. Chem.***, 80, 2568, (1976).

[46] D. E. Goldberg, Genetic Algorithms and Simulated Annealing: An overview., in L. Davis, Eds., "Genetic Algorithms and Simulated Annealing", p 74-88, London Pitmann, 1987.

[47] M. D. Vose and G. E. Liepins, "Punctuated Equilibria in Genetic Search", ***Complex Systems***, 5, 31-44, (1991).

[48] J. H. Holland, Genetic Algorithms and Adaptation, in O. G. Selfridge, E. L. Rissland, and M. A. Arib, Eds., "Proceedings of the NATO Advanced Research Institute on Adaptive Control of Ill-Defined Systems", p 317-333, Plenum Press, New York, 1984.

[49] A. Nix and M. D. Vose, "Modeling Genetic Algorithms with Markov Chains", ***Annals of Mathematics and Artificial Intelligence***, 5, 79-88, (1992).

[50] M. D. Vose, "Modeling Simple Genetic Algorithms", in L. Darrell Whitley, Ed., ***Foundations of Genetic Algorithms***, 2, p 63-73, Aug., 1992.

[51] G. Rudolf, "Convergence Analysis of Canonical Genetic Algorithms", ***IEEE Trans. Neural Networks***, Special Issue on Evolutionary Computing, Vol 5, 1, 96-101, Jan., (1994).

[52] M. Iosifescu, "Finite Markov Processes and Their Applications", Wiley, Chichester, 1980.

[53] H. Muhlenbein, How Genetic Algorithms Really Work I: Mutation and Hill Climbing, in R. Manner and B. Manderick, "Parallel Problem Solving from Nature", 2, p 15-25, Elsevier Science Publishers, Amsterdan, 1992.

[54] H. Aytug, S. Bhattacharya, G. J. Koehler, "A Markov Chain Analysis of Genetic Algorithms with Power of 2 Cardinality Alphabets", **European Journal of Operational Research**, 96, 195, (1996).

[55] J. Suzuki, "A Markov Chain Analysis of Simple Genetic Algorithms", **IEEE Trnasactions on Systems, Man, and Cybernetics**, 25, 4, 655, (1995).

[56] N. G. Gayyord, "Copolymers, Polyblends and Composites", ***Adv. Chem. Ser.***, 76, 142, 1975.

[57] P. A. Small, "Some Factors Affecting the Solubility of Polymers", ***J. Appl. Chem.***, 3, 71, (1953).

[58] L. A. Utracki, "Polymer Alloys and Blends", Hanser Publishers, Munich, 1989.

Table 1**:** List of polymers in polymer database from which data is used to form the 440 input patterns.

1.Main-chain Acyclic Carbon Polymers:
- 1.1)Poly(dienes) (4)
- 1.2)Poly(alkenes) (7)
- 1.3)Poly(acrylic acid) (11)
- 1.4)Poly(acrylic acid ester) (9)
- 1.5)Poly(acrylamides) (2)
- 1.6)Poly(methacrylic acid) (4)
- 1.7)Poly(methacrylic acid ester) (4)
- 1.8)Poly(methacrylamides) (2)
- 1.9)Poly(vinyl ether) (2)
- 1.10)Poly(vinyl thioether) (1)
- 1.11)Poly(vinyl alcohol) (5)
- 1.12)Poly(vinyl ketones) (2)
- 1.13)Poly(vinyl halides) (5)
- 1.14)Poly(vinyl nitriles) (5)
- 1.15)Poly(vinyl esters) (25)
- 1.16)Poly(styrenes) (48)

2.Main-chain Carbocyclic Polymers:
- 2.1)Poly(phenylenes) (19)

3.Main-chain Acyclic Hetroatom Polymers:
- 3.1)Main-chain -C-O-C- Polymers:
  - 3.1.1)Poly(oxides) (22)
  - 3.1.2)Poly(carbonates) (46)
  - 3.1.3)Poly(esters) (70)
  - 3.1.4)Poly(anhydrides) (1)
  - 3.1.5)Poly(urethanes) (32)
- 3.2)Main-chain O-Heteroatom Polymer:
  - 3.2.1)Poly(sulphonates) (1)
  - 3.2.2)Poly(siloxanes) (15)
- 3.3)Main-chain -C-(S)-C- and -C-S-N- Polymers:
  - 3.3.1)Poly(sulfides) (8)
  - 3.3.2)Poly(thioesters) (1)
  - 3.3.3)Poly(sulphones) (2)
  - 3.3.4)Poly(sulphonamides) (2)
- 3.4)Main-chain -C-N-C- Polymers:
  - 3.4.1)Poly(amides) (63)
  - 3.4.2)Poly(imines) (3)
  - 3.4.3)Poly(ureas) (4)
  - 3.4.4)Poly(phosphazenes) (4)
  - 3.4.5)Poly(silanes) (1)
  - 3.4.6)Poly(silazanes) (1)

4.Main-chain Heterocyclic Polymers:
- 4.1)Poly(acetals) (3)
- 4.2)Poly(carboranes) (2)
- 4.3)Poly(piperazines) (2)
- 4.4)Poly(oxadiazoles) (2)

Fig 1: Error plots for variation of number of hidden nodes in a single hidden layer backpropagation neural network.

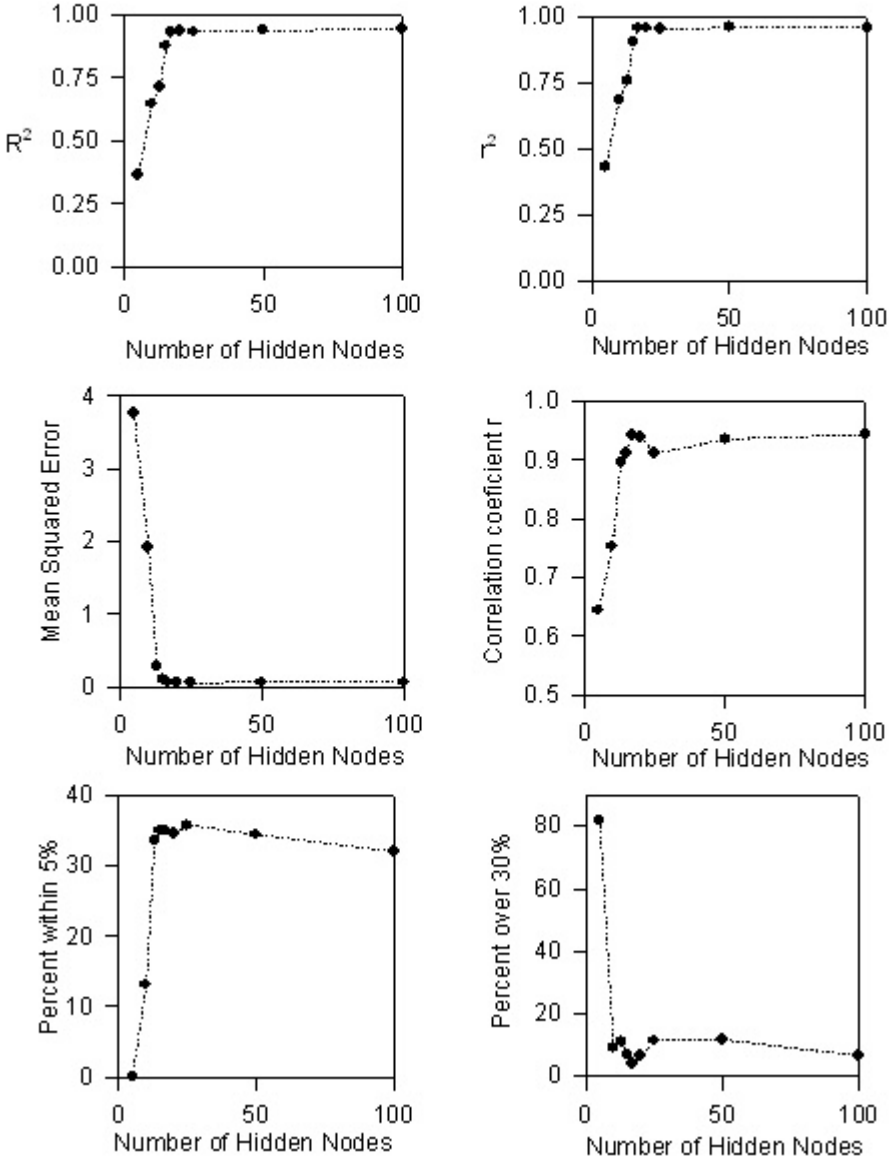


Fig 2: Effect of varying the initial weights.

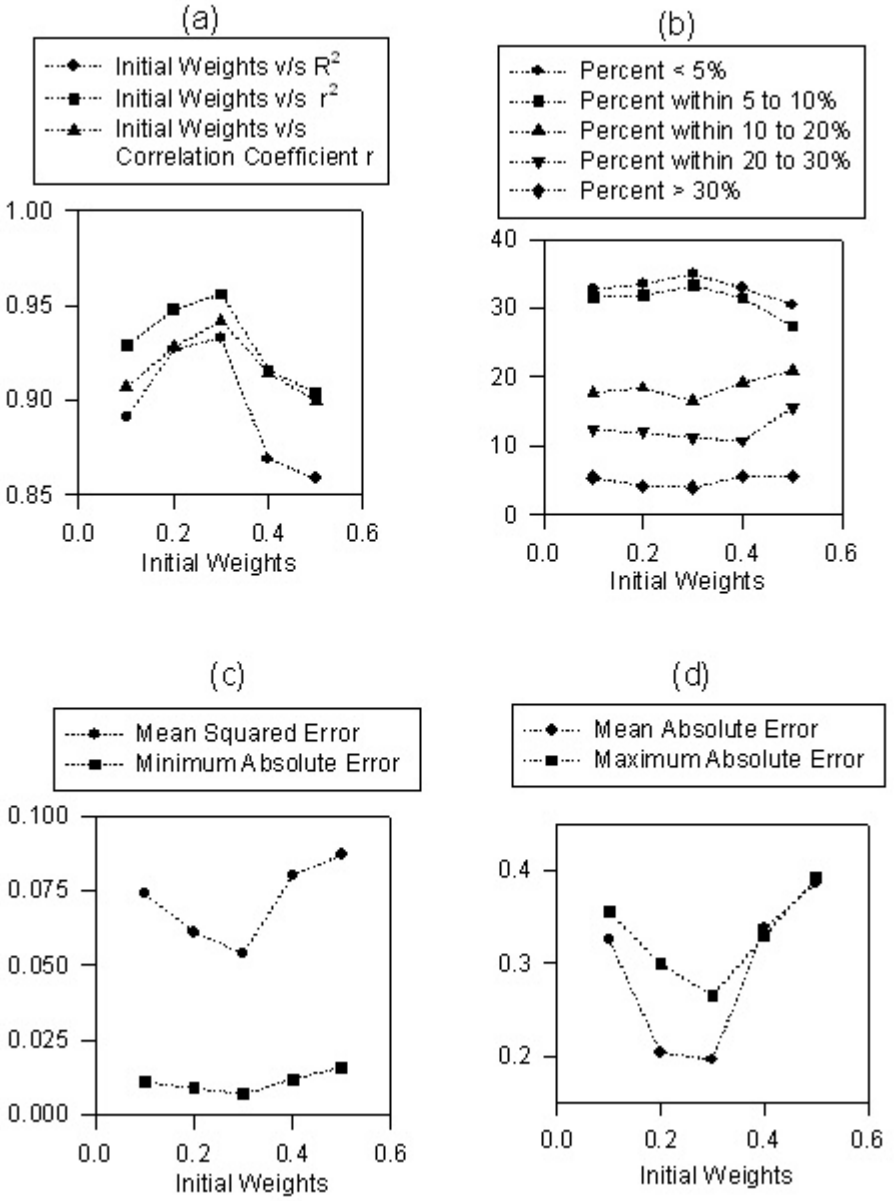


Fig 3: Effect of varying momentum

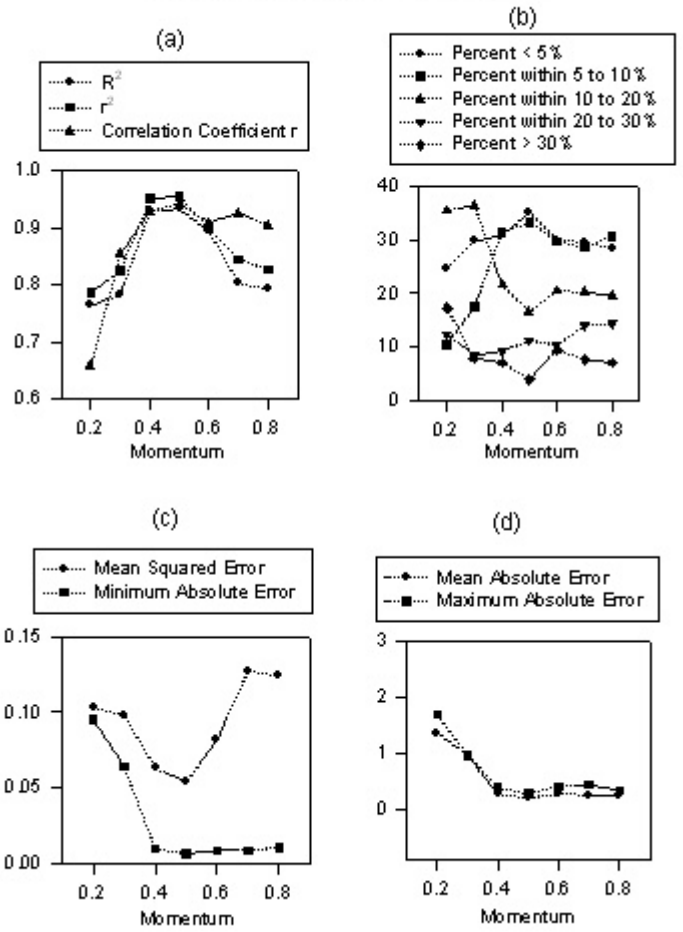


Fig 4: Effect of varying learning rate

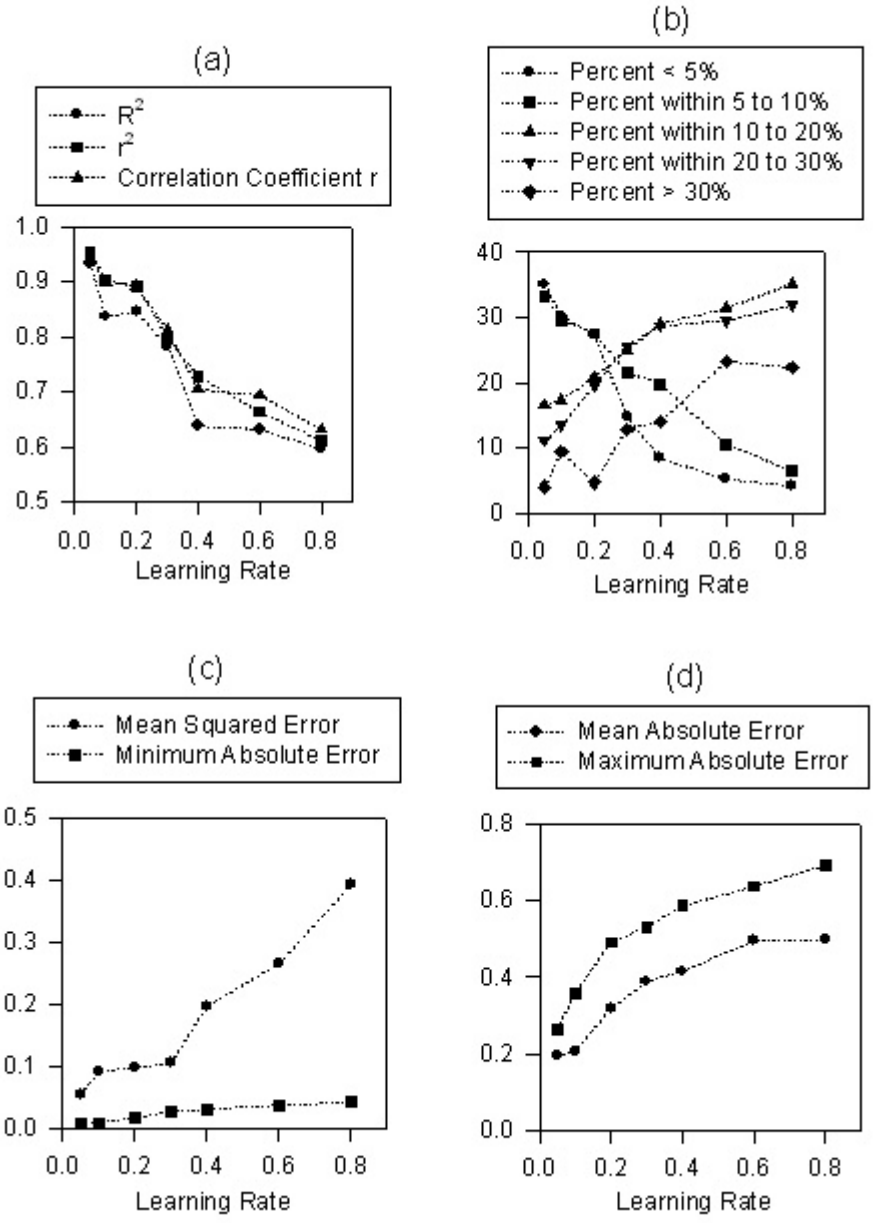


Fig 5: Actual v/s Predicted Output when final model is applied to 88 pattern production file.

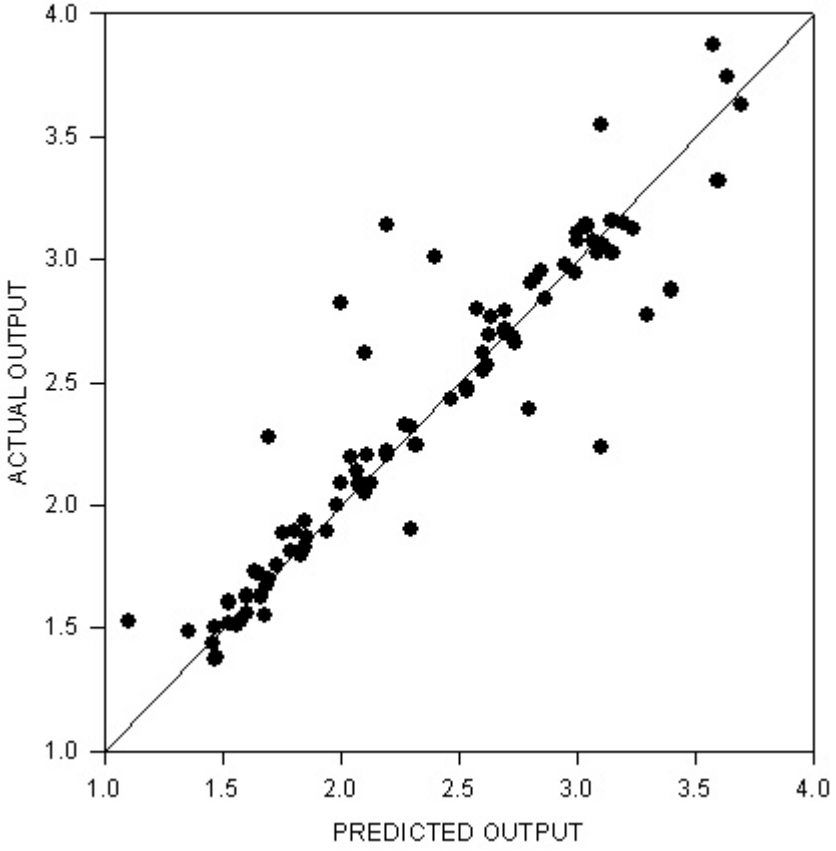


Table 2: Effect of adding a second layer to the standard backpropagation single hidden layer neural network. All activation functions are logistic. Input scaling function is logistic. Number of hidden nodes in first hidden layer = 17. Learning Rate = 0.05. Momentum = 0.5. Initial Weights = 0.3.

Number of Hidden Nodes In Second Hidden Layer	5	10	15
R^2	0.854	0.919	0.893
r^2	0.913	0.922	0.919
Mean Squared Error	0.072	0.068	0.069
Mean Absolute Error	0.285	0.226	0.225
Min. Absolute Error	0.008	0.007	0.007
Max. Absolute Error	0.300	0.285	0.289
Correlation coefficient	0.924	0.936	0.939
Percent within 5%	32.56	34.58	34.74
Percent within 5% to 10%	32.27	33.04	32.95
Percent within 10% to 20%	17.54	18.59	19.01
Percent within 20% to 30%	8.32	7.38	6.39
Percent over 30%	9.31	6.41	6.91

Table 3: a) Results of applying the final model to 24 modified bisphenol-A polycarbonates.
b) Results of applying the final model to 19 modified poly(2,6-dimethyl-1,4-phenylene oxide)

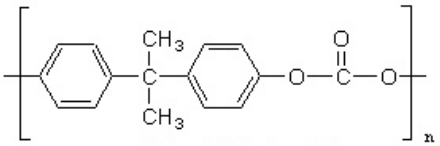
a)

	T_α/T_γ	Dynamic Modulus (20^OC, dynes/cm^2)
Modification PC-1	3.13	5.67×10^9
Modification PC-2	2.70	5.22×10^9
Modification PC-3	2.74	5.38×10^9
Modification PC-4	3.37	6.39×10^9
Modification PC-5	2.58	5.06×10^9

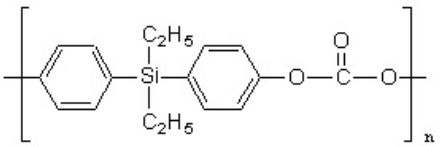
b)

	T_α/T_γ	Dynamic Modulus (20^OC, dynes/cm^2)
Modification PPO-1	1.98	6.32×10^9
Modification PPO-2	2.29	6.59×10^9

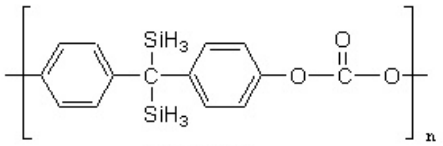
Fig 6



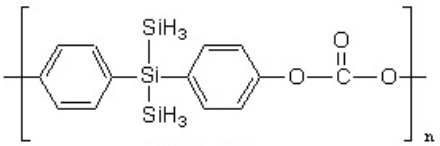
Bisphenol-APolycarbonate(PC)



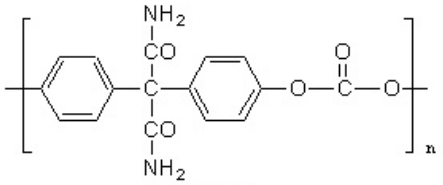
ModificationPC-1



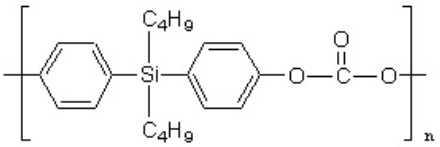
ModificationPC-2



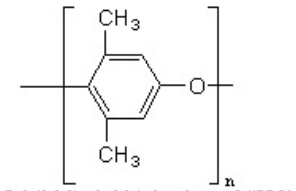
ModificationPC-3



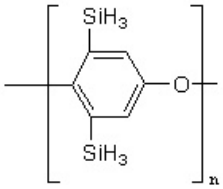
ModificationPC-4



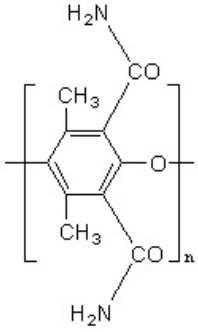
ModificationPC-5



Poly(2,6-dimethyl-1,4-phenyleneoxode)(PPO)



ModificationPPO-1



ModificationPPO-2

Table 4: Values of the interaction energy parameters calculated exactly for different polymers and polymer blend systems using UNIFAC method. Molecular Weight is set at 50,000 and the systems are monodisperse.

Polymer/Polymer Blend System	ζ at 20^{O}C	ξ at 20^{O}C
PC	0.982	0.00721
PS	0.926	0.00796
PPO	0.957	0.00754
PMMA	1.010	0.00792
Modification PC-1	1.231	0.00629
Modification PC-2	1.148	0.00675
Modification PC-3	1.247	0.00735
Modification PC-4	1.482	0.00621
Modification PC-5	1.194	0.00584
Modification PPO-1	0.993	0.00826
Modification PPO-2	0.961	0.00869
PC/PMMA 70/30 Blend	0.991	0.00762
PS/PPO 50/50 Blend	0.938	0.00839
PMMA/Modification PC-1 20/80 Blends	1.146	0.00597
PMMA/Modification PC-2 20/80 Blends	1.129	0.00608
PMMA/Modification PC-3 20/80 Blends	1.155	0.00621
PMMA/Modification PC-4 20/80 Blends	1.327	0.00782
PMMA/Modification PC-5 20/80 Blends	1.083	0.00670
PS/Modification PPO-1 50/50 Blends	0.972	0.00926
PS//Modification PPO-2 50/50 Blends	0.948	0.00984

Table 5: Values of the interaction energy parameters for different polymer blend systems obtained by using GA-Markov chain modeling technique.

Polymer Blend System	ζ at 20^{O}C	ξ at 20^{O}C	Variance in the standard deviation of g for the last 1000 generations in the GA simulation
PC/PMMA 70/30 Blend	0.992 ± 0.002	0.00766 ± 0.00003	0.1086 ± 0.0007
PS/PPO 50/50 Blend	0.935 ± 0.001	0.00834 ± 0.00019	0.1058 ± 0.008
PMMA/Modification PC-1 20/80 Blend	1.148 ± 0.002	0.00602 ± 0.00022	0.1039 ± 0.013
PMMA/Modification PC-2 20/80 Blend	1.127 ± 0.003	0.00605 ± 0.00014	0.1027 ± 0.011
PMMA/Modification PC-3 20/80 Blend	1.153 ± 0.004	0.00620 ± 0.00009	0.1006 ± 0.010
PMMA/Modification PC-4 20/80 Blend	1.322 ± 0.002	0.00801 ± 0.00026	0.0983 ± 0.020
PMMA/Modification PC-5 20/80 Blend	1.080 ± 0.003	0.00713 ± 0.00019	0.1007 ± 0.012
PS/Modification PPO-1 50/50 Blend	0.975 ± 0.001	0.00932 ± 0.00032	0.0986 ± 0.018
PS//Modification PPO-2 50/50 Blend	0.949 ± 0.001	0.00993 ± 0.00029	0.0994 ± 0.015