

COMPUTER AIDED MOLECULAR DESIGN OF SOLVENTS FOR DISTILLATION PROCESSES

Braam van Dyk and Izak Nieuwoudt *

Institute for Thermal Separation Technology, Department of Chemical Engineering,
University of Stellenbosch, South Africa.

ABSTRACT

A Genetic Algorithm based CAMD model was extended to design improved solvents and solvent mixtures for extractive distillation, azeotropic distillation, liquid extraction and liquid chromatography. The algorithm was applied to a number of industrially relevant separation problems. In each case solvents were found that are predicted to perform substantially better than those that are currently used in industry.

Keywords: Solvents, Distillation, CAMD

INTRODUCTION

Separation processes are an integral part of chemical engineering. The purity of a chemical product is among the principal factors influencing its value. Therefore, any method that can increase the purity of a product or decrease the cost of purification will have a direct effect on the profitability of the entire plant.

An important class of separation processes is the solvent-based separations. This includes processes like extractive distillation, liquid-liquid extraction and chromatographic separation. Heterogeneous azeotropic distillation is closely related to these processes. The most important variable in the design of a solvent-based separation process is the choice of solvent.

In the last two decades, several computer aided molecular design methods (CAMD) have seen the light[1-8]. The pros and cons of these methods have been reviewed by van Dyk[9].

A genetic algorithm for the computer-aided molecular design of solvents, called SolvGen, has been developed previously by the authors[10]. In this paper, improvements to the basic algorithm are presented. The algorithm was improved and expanded to include solvent mixtures, liquid-liquid extraction, heterogeneous

azeotropic distillation and chromatographic separations. At the same time the efficiency of the algorithm was improved, resulting in a dramatic speed increase.

The algorithm was applied to a number of industrially relevant separation problems. In each case solvents were found that are predicted to perform substantially better than those that are currently used in industry. A number of these predictions were tested by experiments and found to hold true.

THE APPLICATION OF GENETIC ALGORITHMS TO SOLVENT DESIGN

A detailed description of the application of genetic algorithms in solvent design can be found in references 9 and 10. Only a brief summary of the methodology will be given here.

The Basic Algorithm

The basic genetic algorithm is summarised in Table 1.

Table 0: The Basic Algorithm

1. Initialise a population of chromosomes.
2. Evaluate each chromosome in the population. 2.1. Estimate the properties. 2.2. Calculate the property fitness values. 2.3. Calculate the global fitness.
3. Choose best 10% and copy them to the new generation.
4. Create new chromosomes. 4.1. Choose an operator. 4.2. Choose parent chromosome(s). 4.3. Apply the operator to the parent chromosome(s).
5. Copy the new chromosomes to the new generation until this generation has been filled.
6. Replace the current generation with the new generation.
7. If time is up, go to step 8, else repeat from step 2.
8. Evaluate each chromosome in the population.
9. Return the best solution

Like its biological equivalent the chromosomes are made up of genes. The chromosomes are molecules and the genes are the structural groups that make up the molecule. A population is simply an assembly of molecules. A gene could be a single UNIFAC group or a number of UNIFAC groups that are bonded (pre-defined genes). The pre-defined genes makes it possible to build more complex structures than can be built by linear combination of UNIFAC groups only. It also allows the combination of aliphatic and aromatic compounds. End-genes are defined as groups with only one free bond. Middle genes are defined as groups with two free bonds.

The second step in table 1 is the evaluation of each chromosome. There are several properties that determine the quality of a solvent. The properties and the group contribution methods used to evaluate them are given below:

- Separation factor: Flash calculations using UNIFAC[11] or Modified UNIFAC[12]
- Boiling point and Freezing point: Marrero-Morejon[13] or Joback[14]
- Phase split: From flash calculations
- Recovery of valuable component: Calculated from mass balances

The multiple requirements that must be satisfied must be combined into a single fitness value. This is done by calculating a fitness value for each property and then using a weighted mean as the overall fitness of the chromosome. The weights assigned to each property are determined by the specific problem under consideration and the perceived quality of the group-contribution method for this particular property. For some of the properties a step-function (or Boolean function) is used and for others a sigmoidal function is used[9,10].

Step 4 in table 1 is the creation of new chromosomes. Four genetic operators are used. These are point mutation, one point crossover, insertion and deletion. A Roulette wheel method[9] is used to select an operator, and the chromosome and gene the operator will perform its action on. The use of the insertion and deletion operators entails the insertion or removal of a gene (sub-group) from the chromosome (molecule) and is limited by the restrictions in the structure of a chromosome. Start and end genes may not be inserted or deleted. The maximum number of middle genes is also fixed at six. When applying the crossover operator two new chromosomes are created. This is only allowed if there is enough space for both in the generation that is being constructed. The crossover point is determined randomly within a small interval around the centre of each chromosome. The point mutation operator entails exchanging one gene (sub-group) with another gene (sub-group) from the gene pool (list of available sub-groups)

As was shown by the authors[10], this basic algorithm yields very good results. However, it was clear that this algorithm could be extended to exploit its full potential.

IMPROVING CALCULATION SPEED

In the case of genetic algorithms, a very large part of the processor time is used to generate the pseudorandom numbers used for selection of operators and parent chromosomes. In SolvGen, the computer also spent a lot of time calculating activity coefficients. Improving the speed of these two operations could greatly improve the speed of the entire program.

The Mersenne Twister random number generator[15] was built into SolvGen as a DLL. This random number generator is exceptionally fast, it is equidistant in 623 dimensions and has a cycle length of 2^{19937} . This random number generator is up to 25% faster than the standard generators in compilers and it avoids the lattice problem.

In the UNIFAC method some of the parameters are functions of composition and others not. When evaluating new solvents, the parameters of the components that need to be separated don't change. A typical liquid-liquid case would require five to ten iterations in the flash calculations for the compositions to converge to within

acceptable tolerances. This implies that the UNIFAC calculations would have to be repeated 500,000 to 1,000,000 times in a ten generation run with 10,000 chromosomes! In the case of vapour-liquid calculations where a simple non-iterative bubble point calculation is done, the problem is less severe. Any decrease in the time required calculating the selectivity would be greatly magnified through the number of repetitions of the calculations. Here, it is the separation of the UNIFAC calculations into composition dependent and composition independent parts that allows the improvement in speed. The variables that do not depend on the composition need only be calculated once, at the start of the flash calculation. Only the composition dependant variables need be recalculated at every iteration. The overall increase in speed is remarkable. To perform the flash calculation 100,000 times with these improvements takes only 20% of the time it would have required if it was not implemented.

These improvements made the SolvGen program run up to 5 times faster than it initially ran!

It is also evident that the equilibrium calculation for each of the newly created solvents is completely independent and that all the equilibrium results only need to come together when the solvents are ranked. The Delphi programming environment handles threads easily and efficiently. On a 4-processor machine the total number of solvents may be split into four groups and each of the groups can be run in a thread on a different processor. On machines with less processors, more than one thread will run on the same processor. The threading makes SolvGen almost 4 times faster on a 4-processor machine.

IMPROVING GENE SELECTION

Should a gene contain a functional group that would greatly assist the separation, as well as one that would only slightly counter this effect, the gene as a whole would still assist the separation. Ideally, the selection probability of the gene should be increased. Using the Robbins chart would result in the increase in selection probability due to the first functional group being cancelled by the decrease due to the second group. Clearly, this is not the best method of biasing the selection probabilities.

To solve this problem, a quantitative method was adopted. The activities of the key mixture components are calculated for the case where no solvents are present. Each gene is then added as a solvent on its own and the effect of its presence is calculated by repeating the UNIFAC calculations. This gives a quantitative indication of the effect that this gene as a whole would have on the separation. The adjustment to the selection probability of the gene is then made proportionately to the change in the selectivity it causes. The adjustment is made according to the following formula:

$$SP^{new} = SP^{old} \left(\frac{\alpha_{gene}}{\alpha_{mixture}} + b \right) \quad (1)$$

With

- SP^{new} : The new selection probability
- SP^{old} : The old selection probability
- α_{gene} : The relative volatility of the key components in the presence of the gene
- $\alpha_{mixture}$: The relative volatility of the key components with no solvents present
- b : An adjustable offset

The old selection probabilities, SP^{old} , have values of either 1 or 0, depending on whether the use of the gene is allowed in the solvent design. The offset is an adjustable value that will determine the extent to which the selection probabilities are biased.

IMPROVING OPERATOR SELECTION

To find the optimal set of selection probabilities for all possible mixtures beforehand is impossible. However, it is possible to find the optimal set while the genetic algorithm is running. This is done through a process called automatic tuning.

Automatic tuning requires that the fitness of each new chromosome be compared with that of its parent chromosome. If the child chromosome has a higher fitness than its parent, the selection probability of the operator that created the child is increased by a small amount. If it is lower, the selection probability is decreased.

This process allows the optimal set of selection probabilities for any mixture to be found within a few generations. As the composition of the chromosome population changes, these optimal values may also change. Through automatic tuning, the genetic algorithm will continue to adapt itself to this change in the population.

To illustrate the ability of automatic tuning to find the optimal parameter set four runs were done with the same feed composition and requirements, but with different initial values for the selection probabilities of the genetic operators. The values of these selection probabilities during each generation are shown in the following figures.

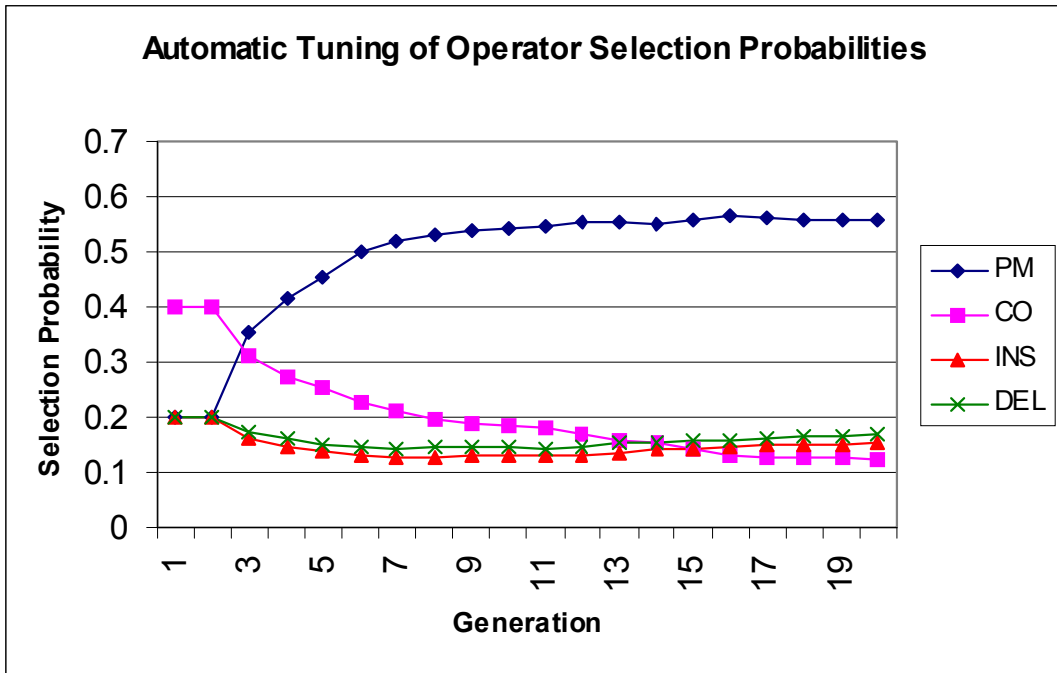


Figure 1: Automatic Tuning of Operator Selection Probabilities

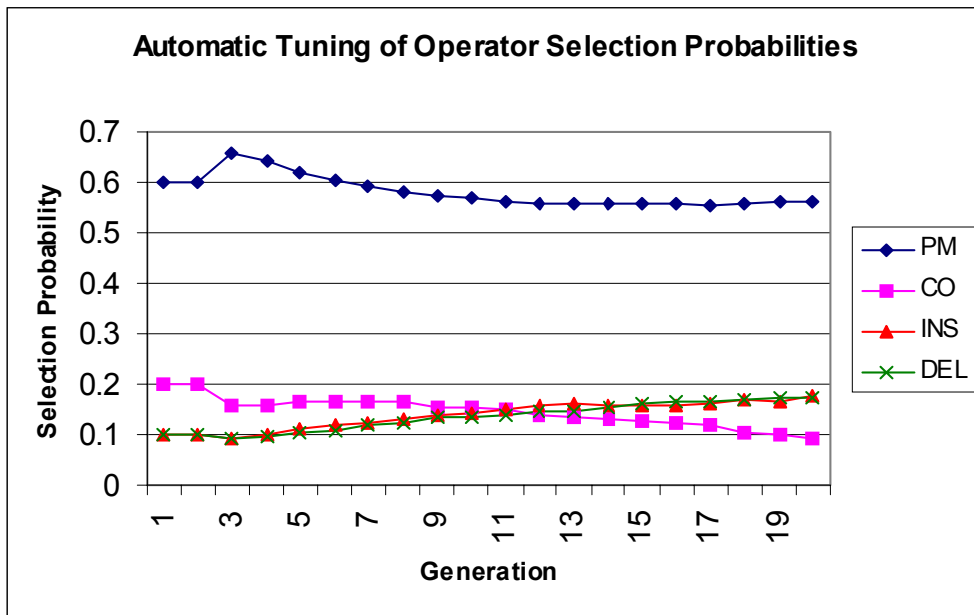


Figure 2: Automatic Tuning of Operator Selection Probabilities

As can be seen from figures 1 and 2, the selection probabilities of the operators quickly converge to approximately the same values, regardless of the starting point. Automatic tuning is only started after the first two generations to allow the system to stabilize first. If this is not done, the point mutation operator will usually completely dominate the other operators. A possible reason for this phenomenon is that the mutated chromosomes of the first few generations greatly outperform those of the initial random generation. This large increase in fitness will then cause a large

increase in the selection probability of the operator and a subsequent decrease in the selection probabilities of the other operators. The effect of automatic operator selection tuning on the overall algorithm performance can be seen in figure 3 below.

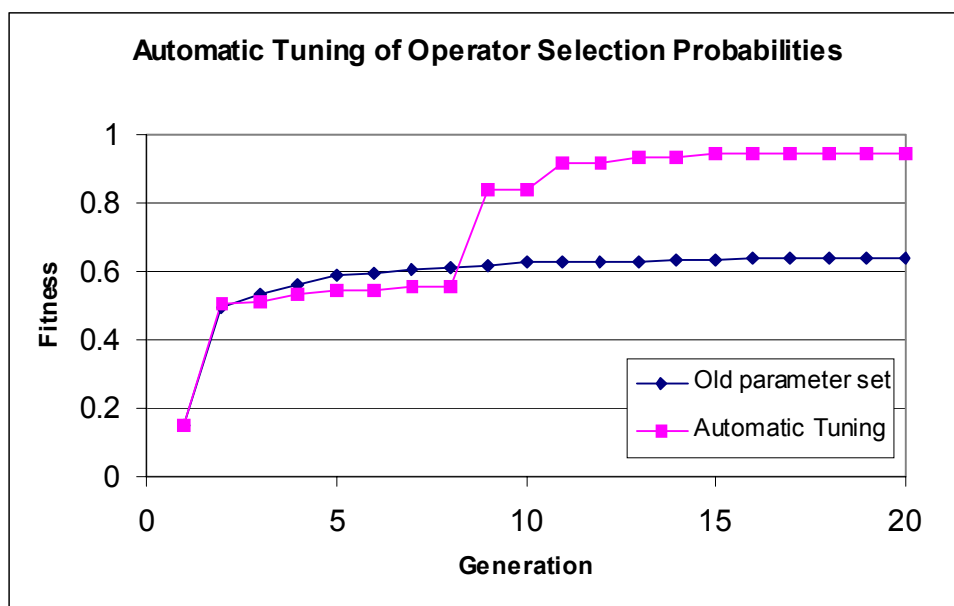


Figure 3: Effect of Automatic Tuning vs. Fixed Selection Probabilities

PHYSICAL VIABILITY OF GENERATED SOLVENTS

Gani et al[4] proposed rules for testing for the viability of a molecule. These rules are aimed at producing molecules that satisfy the octet rule and other physical considerations. In constructing aromatic and other cyclic compounds some of the Gani rules are violated.

The problem of constructing molecules that are not physically viable molecules is to some extent avoided by the use of pre-constructed genes[9,10]. There is however still the possibility of combining two genes that cannot form a feasible molecule.

Generally molecules are unstable if two heteroatoms (i.e. O, N and S) are bonded to the same carbon atom and at least one is also bonded to a hydrogen atom. If neither of the two heteroatoms were bonded to hydrogen atoms, the combination could be stable. Another type of bond that, although may be physically viable, should be avoided is that of one heteroatom to another, e.g. peroxides. These compounds are usually highly reactive and would not normally be considered as solvents.

In order to implement this knowledge into a rule for physical viability, the genes are classified according to the atoms with free bonds. This classification is shown in Table 2.

Table2: Free Bond Classification

Type	Example	Description
I	HO-	Bonding atom is a heteroatom bonded to a hydrogen atom.
II	R-O-	Bonding atom is a heteroatom bonded to a carbon atom.
III	HO-CH ₂ -	Bonding atom is a carbon atom bonded to a heteroatom, which is in turn bonded to a hydrogen atom.
IV	R-O-CH ₂ -	Bonding atom is a carbon atom bonded to a heteroatom, which is in turn bonded to a carbon atom.
V	R-CH ₂	Bonding atom is a carbon atom bonded to another carbon atom.

Table 3: Allowed Combinations of Genes

	I	II	III	IV	V
I	✗	✗	✗	✗	✓
II	✗	✗	✗	✓	✓
III	✗	✗	✓	✓	✓
IV	✗	✓	✓	✓	✓
V	✓	✓	✓	✓	✓

The allowable combinations in Table 3 can be summarized by a single inequality:

$$N_5 + N_4 \geq N_2 + N_1 \quad (2)$$

With

N_i : The number of genes of type i in the chromosome

In order to eliminate structures that do not comply with these rules, a penalty system in equation 3 is used. The fitness of a chromosome is penalised according to equation 3.

$$P = P_v [(N_1 + N_2) - (N_4 + N_5)] \quad (3)$$

With

P : The penalty value

P_v : Scaling constant

This simple rule for physical viability, in combination with the pre-constructed genes, allows practically all physically improbable structures to be eliminated. The result is a final generation of candidate solvents that do not only meet all the requirements in terms of physical properties, but are also physically viable molecules. The quality of the results produced by the SolvGen algorithm is greatly improved in this manner.

BLENDED SOLVENTS

Encoding Scheme

In order to extend the SolvGen algorithm to design blended solvents; new data structures are required. A superchromosome consisting of up to 4 chromosomes (molecules) is defined as depicted in figure 4.

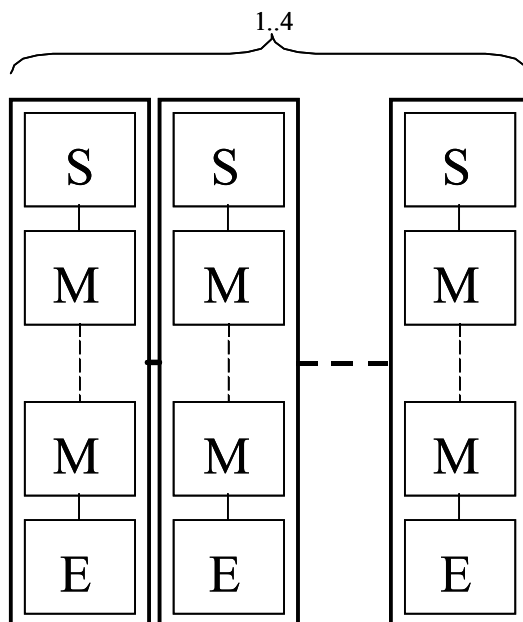


Figure 4: A Superchromosome

The superchromosome is similar to the chromosomes used in single solvent design. Instead of being constructed from genes, the superchromosome is constructed from supergenes. Each supergene is in fact a chromosome in itself.

The number of supergenes is allowed to vary between one and a specified maximum. In this study the maximum number of supergenes was arbitrarily set to four. The composition of the blended solvent is taken to be equal parts of each of the individual solvents in the blend. This means that the concentrations of the solvents in the blend may be established in discrete intervals.

Reproduction Scheme

New operators had to be defined that operate directly on the superchromosomes. The existing operators (point mutation, crossover, insertion and deletion) will act upon the supergenes/chromosomes that comprise the superchromosome.

These new operators are listed in Table 4, along with the normal genetic operators used in the single solvent version of SolvGen.

Table 4: Genetic Operators

Single Solvent Operator	Description	Blended Solvent Operator	Description
Insertion	Insert a random gene into the chromosome.	Super Insertion	Insert a complete molecule, selected from the entire population of existing molecules, into the superchromosome.
Deletion	Delete a random gene from the chromosome.	Super Deletion	Delete a random molecule from the superchromosome.
Point Mutation	Replace a gene in the chromosome with a randomly selected gene from the pool of available genes.	Super Point Mutation	Replace a complete molecule in the superchromosome with one selected from the entire population of existing molecules.
Crossover	Recombine two parent chromosomes to create two new chromosomes. Each new chromosome consists of half of each parent.	Super Crossover	Recombine two parent superchromosomes to create two new superchromosomes. Each new superchromosome consists of half of each parent.

The selection of operators is done using the same Roulette Wheel method as in the single solvent version of the SolvGen Algorithm. The principal change in the algorithm from single to multiple solvents is the initialisation of the first generation of superchromosomes.

Simply initialising four random chromosomes in each superchromosome yielded very poor results. The algorithm repeatedly failed to find solvents that meet the requirements within a reasonable number of generations. This is understandable, as the search space has increased by a power of four, while the number of individuals in the population stayed constant.

Increasing the size of the population to solve this problem is not desirable, as this will slow down the search in direct proportion to the number of extra individuals.

Symbiosis

In the symbiosis method the individual chromosomes within a superchromosome are evolved as complete molecules from the very first step. Each of these molecules is measured against all of the molecular property requirements (boiling points etc.) while the superchromosome as a whole is measured against the selectivity requirement (and phase split, if required).

The symbiosis method comprises of two parts: the initialisation stage and the evolution stage. The initialisation stage starts with each superchromosome being initialised to a single random molecule. A number of generations are then allowed for these solvents to evolve.

Molecules are then added one at a time and also allowed a number of generations to evolve until the maximum number of molecules in a blend has been reached. The components that are already present in the blend are kept fixed during these generations. This allows each new solvent that is added to evolve to a structure that

will aid the solvents already present. In this stage only the single solvent genetic operators are used.

In the second phase, all the solvents in the blend are allowed to evolve and the entire set of genetic operators listed in Table 4 are used.

The method is best explained via an example. Consider the case where a maximum of four solvents is allowed in the blend. The total population size is 10,000 superchromosomes. The process is diagrammatically depicted in Figure 5.

The calculations can be summarised step-wise:

Step 1

A single solvent design is run for a specified number of generations. Upon completion, the 10 best unique solvents from the population of 10,000 are selected.

Step 2

Copies of the solvents selected in step 1 are paired up with 1000 new, randomly generated solvents each. This results in a population of 10,000 binary mixtures in 10 groups of 1000 each. Every individual in a specific group has the same first solvent.

Step 3

The first solvent in each of the mixtures created in step 2 is kept fixed while the second solvents are allowed to evolve for a number of generations. Upon completion, the ten best unique mixtures from each of the ten groups created in step 2 are selected.

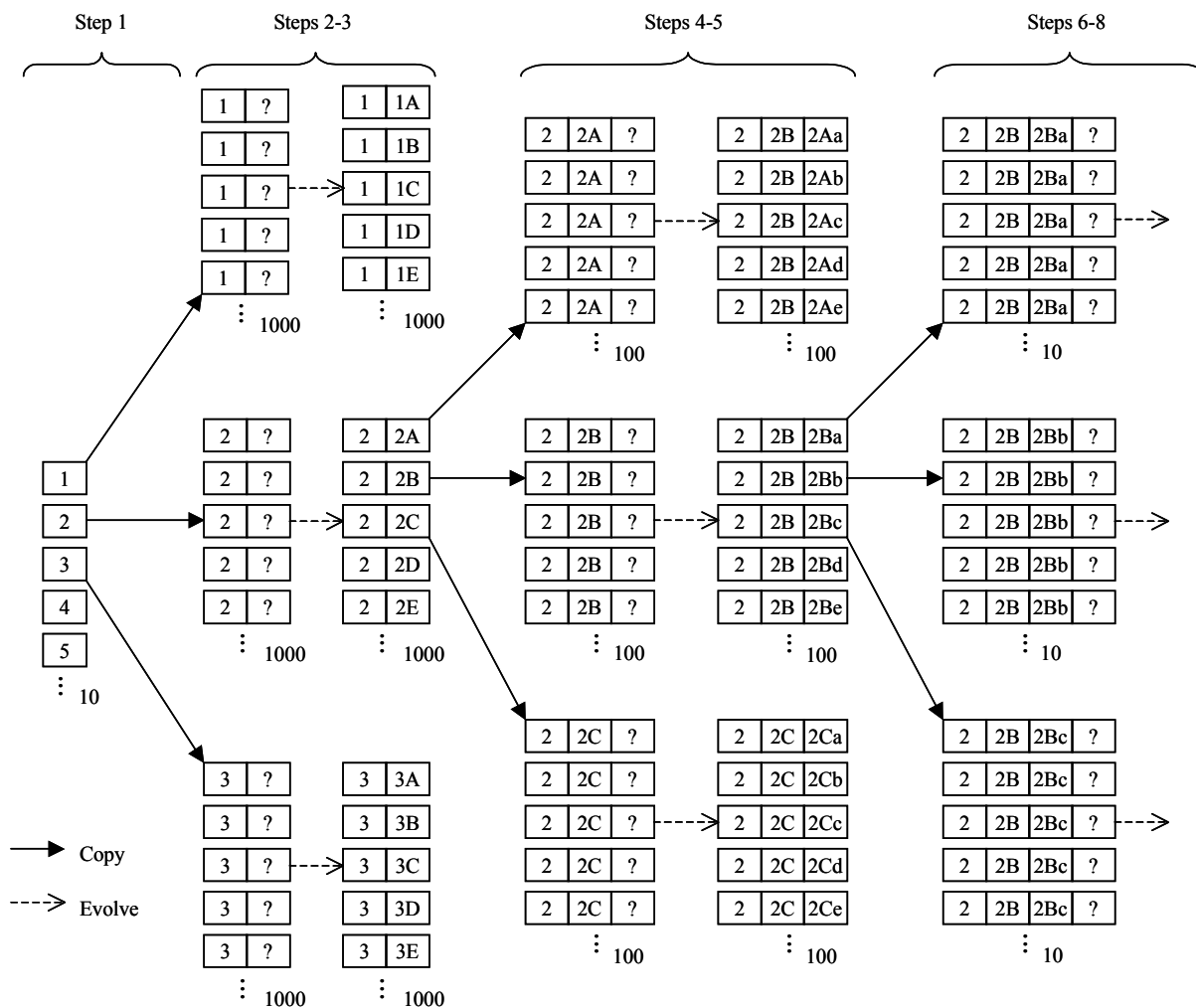


Figure 5: The Symbiosis method

Step 4

Copies of the binary mixtures selected in step 3 are now combined with 100 randomly generated solvents each. This results in a population of 10,000 ternary mixtures in 100 groups of 100 each. Each member of a specific group has the same first and second solvents.

Step 5

The first and second solvents in each mixture are kept fixed while the third solvents are allowed to evolve. Upon completion, the ten best unique mixtures from each of the 100 groups created in step 4 are selected.

Step 6

Copies of the ternary mixtures selected in step 5 are combined with 10 randomly generated solvents each. This results in a population of 10,000 quaternary mixtures, divided into 1000 groups of ten each. Each member of a group has the same first, second and third solvents.

Step 7

The first three solvents in each blend are kept fixed while the newly added fourth solvents are allowed to evolve for a number of generations.

Step 8

After completion of step 7 above, all of the solvents in each blend are allowed to evolve freely. All of the operators in Table 1 are used and so the number of solvents in each blend may vary due to the super insert and super delete operations.

As the symbiosis method works with complete molecules, the fitness is always measured against all of the required properties. Each molecule in the blend must meet the requirements for the boiling point and freezing point. The blend as a whole must meet the requirement for selectivity and recovery (and phase split, if required).

The weighted sum of these property fitness values is assigned to the superchromosome. For properties like boiling and freezing points, the property fitness of the blend is taken to be the average of the individual property fitness values of the molecules in the blend. The selectivity fitness is that of the superchromosome (solvent blend).

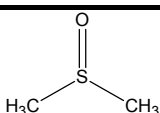
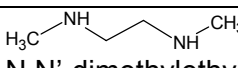
EXTRACTIVE DISTILLATION CASE STUDIES

Acetone / Methanol

Ethylene Glycol or Water may be used as extractive distillation solvent to recover acetone as distillate.

Improved solvents have previously been generated for this system[10]. The diamine performance was experimentally verified. The ability of blended solvents to outperform the pure solvents was investigated. Two solvent blends were designed. The relative volatilities attained with the individual solvents are also listed for comparison.

Table 5: Blended Solvents for the Acetone (1) / Methanol (2) System

Pure solvents			Blended Solvents	
Solvent	α_{12}	T_b [K]	Solvent	α_{12}
 Dimethyl sulfoxide (DMSO)	2.98*	464.1	DMSO +DMEDA	8.14*
 N,N'-dimethylethylenediamine (DMEDA)	5.81	405.9		

* α_{12} -value from UNIFAC

As can be seen from the listed relative volatilities, the blended solvent is predicted to perform better than their individual components. The accuracy of these predictions, and thus the existence of this synergy, must still be verified experimentally.

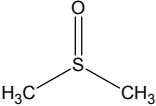
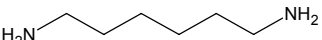
In both cases the blend was made up from DMSO and a secondary amine. Primary amines were not allowed, as these will react with the acetone.

Methyl Acetate / Methanol

This system also forms a minimum boiling azeotrope, which may be separated using ethylene glycol monomethyl ether (2-methoxyethanol) as solvent. The estimated relative volatility with this solvent is 2.21 (UNIFAC) with methyl acetate in the distillate. As with the previous system, improved solvents have been designed for this system [10]. In this work, the focus was moved to solvent blends.

The solvent blend proposed for this system is given in Table 6. The effects of the individual components are again listed for comparison.

Table 6: Blended Solvents for the Methyl Acetate (1) / Methanol System

Pure solvents			Blended Solvents	
Solvent	α_{12}	T_b [K]	Solvent	α_{12}
 Dimethyl sulfoxide (DMSO)	3.55*	464.2	DMSO + Hexamethylenediamine	7.83*
 Hexamethylenediamine	2.80	472.8		

* α_{12} -value from UNIFAC (Hansen et al, 1991)

The synergistic effect of the blend is again evident in these results. The blended solvent attains a much higher relative volatility than was the case with the individual components. The relative volatility with the diamine was experimentally confirmed. Again, this prediction of the blended solvent should be verified experimentally.

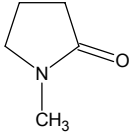
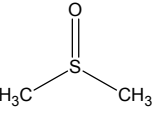
Ethyl Acetate / Ethanol

The third system is that of ethanol and ethyl acetate. This system forms a low boiling azeotrope. Aromatics, like trimethylbenzene, have been proposed as solvents for separating this system with extractive distillation, resulting in ethanol being recovered in the distillate. The predicted relative volatility for this separation is 3.27 (UNIFAC). Improved pure solvents for this separation have also been designed previously[10].

The ability to manipulate which component is recovered in the distillate will give greater flexibility in the design of separation systems. It was therefore proposed to

find a blended solvent that would allow the recovery of ethyl acetate in the distillate. The designed solvents are given in Table 7. The relative volatilities were estimated with standard UNIFAC (Hansen et al, 1991).

Table 7: Blended Solvents for the Ethyl Acetate (1) / Ethanol (2) System

Pure solvents			Blended Solvents	
Solvent	α_{12}	T_b [K]	Solvent	α_{12}
 1-Methyl-2-pyrrolidone (NMP)	3.53*	475.2	NMP + DMSO	5.92*
 Dimethyl sulfoxide (DMSO)	4.56*	464.2		

* α_{12} -value from UNIFAC (Hansen et al, 1991)

Both of the individual solvents are able to reverse the relative volatility of the system. Using them together results in an even higher relative volatility, due to a synergistic interaction. This prediction should be verified experimentally.

In the three systems for which blended solvents were designed, it was possible in each case to find a solvent blend that outperformed its individual components. This is not always the case. Much more often the relative volatility attained by the blend is approximately the average of that of the individual components. When the requirement for the relative volatility was increased in these cases, the algorithm converged to a single solvent.

DESIGNING ENTRAINERS FOR HETEROGENOUS AZEOTROPIC DISTILLATION

Residue Curve Tracing Method

Residue curves are used extensively to explain phenomena in azeotropic distillation. As all residue curve lines originate in negative nodes, we need only trace these lines back to their origin to find any suitable heterogeneous azeotrope present, provided that we start within the correct distillation region.

As binary azeotropes may also form negative nodes, this method will locate both binary and ternary heterogeneous azeotropes that are the lowest boiling species in their distillation regions. Should a heterogeneous binary azeotrope form that boils at a lower temperature than any other pure component or azeotrope, it may also be used to affect the desired separation, provided it forms between the entrainer and one of the key components.

The process consists of a series of liquid-liquid flashes and bubble point calculations to find the vapour composition and pressure. The vapour composition calculated in such a flash is then used as the feed composition to the next flash. In this manner the residue curves are traced back to their origin. The azeotrope is found within a specified tolerance by doing a convergence check on the vapour composition calculated in each step.

To ensure that lowest boiling azeotrope is found, several starting points may be used as shown in Figure 6. Should the residue curve map be divided by distillation boundaries, at least one of these starting points should be in the correct distillation region to terminate at the lowest boiling azeotrope. Figure 6 also shows the stepwise progression from one of these starting points to the azeotrope.

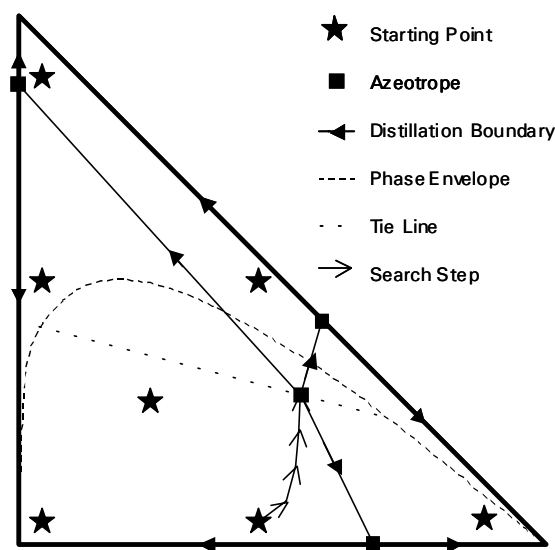


Figure 6: The Residue Curve Tracing Method

The speed of the method for locating ternary azeotropes may be further increased by making use of the fact that in ternary systems, the ternary azeotrope must be connected by a distillation boundary to a binary saddle. Instead of the seven starting point shown in Figure 6, one need only start at the binary azeotropes. This typically gives a maximum of three starting points instead of seven.

The method may also be applied to quaternary and higher mixtures. The search may be done in the multi-component composition space, but as quaternary azeotropes are exceedingly rare and the very existence of quinary azeotropes a point of debate[17].

The search method should not normally terminate at saddle azeotropes unless a search step ends at exactly the azeotropic composition. Because residue curves both start and end at saddle points, the tracing algorithm should change course near a saddle azeotrope to follow residue curves that do not terminate at the saddle. This is illustrated in Figure 7.

If the presence of a saddle point azeotrope is suspected, the multiple starting point method discussed above should be used. If the binary azeotropes are used as

starting points, the tracing algorithm may run along a residue curve that terminates at the ternary saddle.

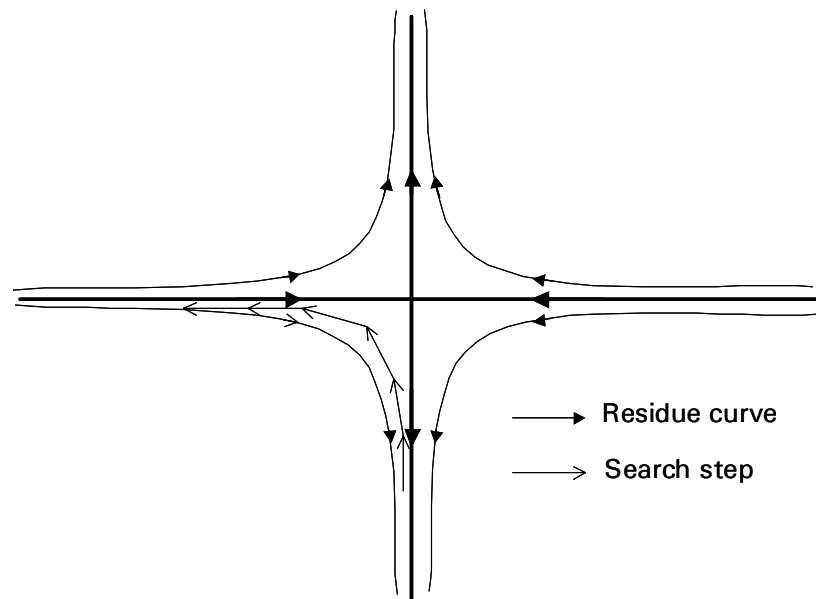


Figure7: The Residue Curve Tracing Method Near Saddle Azeotropes

As heterogeneous azeotropes can only be saddles or negative nodes[17], this method should always locate a suitable azeotrope if one is present in the system. However, as the ternary azeotrope, if one is present, will not necessarily be the only negative node present in the system, the end-point of each search should be evaluated to ensure that it is indeed a suitable azeotrope.

The method is extremely efficient. The specific number of iterations required will depend on the system under consideration.

A Fitness Function for Entrainers

The first step in assigning a fitness value to a candidate entrainer is to determine whether a suitable heterogeneous azeotrope is formed by the addition of the entrainer. This azeotrope may be ternary, as is the case when benzene is added to a mixture of water and ethanol, or it may be binary as is found when ethyl acetate is added to a mixture of water and acetic acid.

When more than one suitable azeotrope is formed, and both are negative nodes in their respective distillation regions, the composition of the feed will determine which azeotrope will form the distillate. If it is possible to manipulate the feed composition to fall into the more beneficial distillation region, the azeotrope that forms the negative node in that region may be considered.

When there are components present in the feed that do not form part of the azeotrope, the effect of their presence on the azeotrope should be determined as was discussed above.

Once it has been determined that a suitable heterogeneous azeotrope is formed by the addition of the entrainer and that this azeotrope is a negative node, a relative

fitness must be allocated to the chromosome. As stated previously, we would like the miscibility gap at the azeotropic point to be as wide as possible to increase the ease of separation. As such, the liquid-liquid separation factor at the ternary azeotrope may serve as an indication of the suitability of the entrainer.

The separation factor, β_{ij} , is similar to the relative volatility, α_{ij} , but defined for liquid-liquid systems, instead of vapour-liquid systems. The definition of the separation factor is given in Equation 4.

$$\beta_{ij} = \frac{\frac{x_i^I}{x_i^{II}}}{\frac{x_j^I}{x_j^{II}}} \quad (4)$$

With

- β_{ij} : The separation factor of components i and j
- x_i^I : The mole fraction of component i in liquid phase I
- x_i^{II} : The mole fraction of component i in liquid phase II

The separation factor is used in the fitness function in the place of the relative volatility. The penalty function used to help force a liquid-liquid split in the design of solvents for liquid-liquid extraction[9], may also be used.

The fitness function developed here may be improved in one more manner. It may well be possible that an azeotrope forms that consists of mostly entrainer. Although such an entrainer may have a very high selectivity, it will not be an economically attractive choice, due to the large amount of entrainer that would have to be evaporated and subsequently condensed. The algorithm may be encouraged to find more suitable entrainers by including a penalty based on the amount of entrainer present in the azeotrope.

To implement this penalty system, the selectivity is multiplied by a scaling factor, as shown in Equation 5.

$$\beta_{12}^* = \beta_{12} \frac{c_1}{1 + c_2 \exp(x_E - 0.5)} \quad (5)$$

With

- β_{12}^* : The scaled selectivity
- x_E : The mole fraction entrainer in the azeotrope
- c_1, c_2 : Constants

The severity of the penalty may be adjusted by modifying the constants. The scaling factor also rewards entrainers with small molar fractions in the azeotrope.

With the inclusion of this scaling factor, the search algorithm will not only find highly selective entrainers, but also those with high capacities, i.e. entrainers that make up a smaller part of the azeotropic composition. This should lead to less expensive separations processes, increasing the profitability of the entire plant.

HETEROGENOUS AZEOTROPIC DISTILLATION CASE STUDIES

In this section, the ability of the SolvGen algorithm to find suitable entrainers to separate binary systems with heterogeneous azeotropic distillation will be demonstrated. The entrainers used industrially to form heterogeneous azeotropes are also given.

In the calculations used to locate azeotropes, the accuracy of the vapour pressures of the pure components is much more important than with the other separation process discussed here. The method currently used in SolvGen, fits a simplified version of the Antoine equation through the estimated boiling and freezing point data. This is sufficiently accurate near the boiling point of the component, but further away from the normal boiling point the accuracy is variable.

The magnitude of this problem is illustrated by the ethanol/ benzene/ water azeotrope. The composition of this azeotrope from the literature (Gmehling et al, 1994) is compared with that predicted by SolvGen (using UNIFAC) in Table 8.

Table 8: Comparison of Azeotropic Compositions, 338K

Literature Composition		Predicted Composition	
Ethanol	0.2281	Ethanol	0.2806
Benzene	0.5387	Benzene	0.5200
Water	0.2332	Water	0.1994


This lack of accurate group-contribution methods is the Achilles Heel of all CAMD methods. Even the most advanced design algorithm can only function in the search space provided by the group-contribution methods it employs.

The effect of the problem may be controlled by setting the boiling point requirements for the entrainer to a narrow band around the working temperature. The algorithm should then be run with different temperatures, to find all possible entrainers. By confining the entrainer to a narrow boiling point range, it is also confined to the range of highest accuracy in the vapour pressure estimation.

Ethanol / Water

The dehydration of ethanol with benzene as the entrainer in a heterogeneous azeotropic distillation process is a classic textbook example. As benzene is a suspected carcinogenic, the possibility of using alternative entrainers should be investigated. The azeotropic composition of the ethanol / water / benzene system (predicted with UNIFAC) is given in Table 9 and for the proposed entrainers, in Table 10.

Table 9: Industrial Entrainers for the Ethanol / Water System (350K)

Benzene				
				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Ethanol	0.3226	0.4658	0.1854	7.75
Water	0.2308	0.4479	0.0230	θ
Entrainer	0.4465	0.0863	0.7916	0.511

The β_{21} value is the selectivity, calculated between the two liquid phases using Equation 4. θ is the liquid phase ratio and may be calculated with

$$\theta = \frac{L_2}{L_1 + L_2} \quad (6)$$

With

θ : The phase ratio

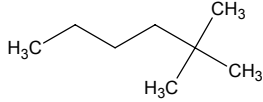
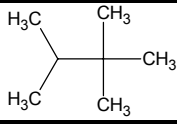
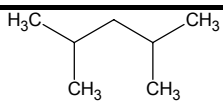
L_j : The total molar amount of liquid phase j

The selectivity is only one of the indicators of the suitability of an entrainer. A good entrainer must not only have a high selectivity, but also a high capacity. This implies that the molar fraction of the entrainer in the azeotrope must be as small as possible. Furthermore, the separation between the two liquid phase compositions should be as wide as possible.

If the entrainer constitutes a large percentage of the azeotrope, a lot of energy will be spent evaporating the entrainer. A larger diameter column will also be required due to the high flow rates caused by the large amount of entrainer in the column. Thus, both the capital and running cost of the process will be increased.

All the azeotropic compositions were calculated with UNIFAC at 350K and the bubble point pressure of the mixture.

Table 10: Entrainers for the Ethanol / Water System (350K)

 2,2-Dimethylhexane				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Ethanol	0.4466	0.6468	0.1164	7.78
Water	0.1821	0.2886	0.0067	θ
Entrainer	0.3713	0.0645	0.8769	0.377
 2,2,3-Trimethylbutane				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Ethanol	0.3235	0.5903	0.1017	8.62
Water	0.1587	0.3410	0.0068	θ
Entrainer	0.5187	0.0687	0.8915	0.546
 2,4-Dimethylpentane				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Ethanol	0.3412	0.6019	0.0992	8.70
Water	0.1615	0.3287	0.0062	θ
Entrainer	0.4973	0.0694	0.8946	0.519

The 2,2-dimethylhexane has a selectivity only marginally higher than benzene, but comprises a significantly smaller portion of the azeotrope. The 2,2,3-trimethylbutane and 2,4-dimethylpentane each make up a larger fraction of their respective azeotropic compositions, but have much better selectivities than benzene. Which of these two factors carries the most weight will have to be determined by a more detailed study of the process economics.

Very importantly, all three of the proposed entrainers are completely non-toxic, while benzene is both toxic and a suspected carcinogenic. This weighs heavily in favour of the proposed entrainers.

The residue curve maps for the systems ethanol / water / benzene and ethanol / water / 2,2-dimethylhexane are shown in Figure 8 and Figure 9. These figures show that the same separation train design may be used with both entrainers. However, the separation between the compositions of the aqueous and organic phases formed in the decanter are significantly wider for 2,2-dimethylhexane, as indicated by the length of the tie-line in Figure 9.

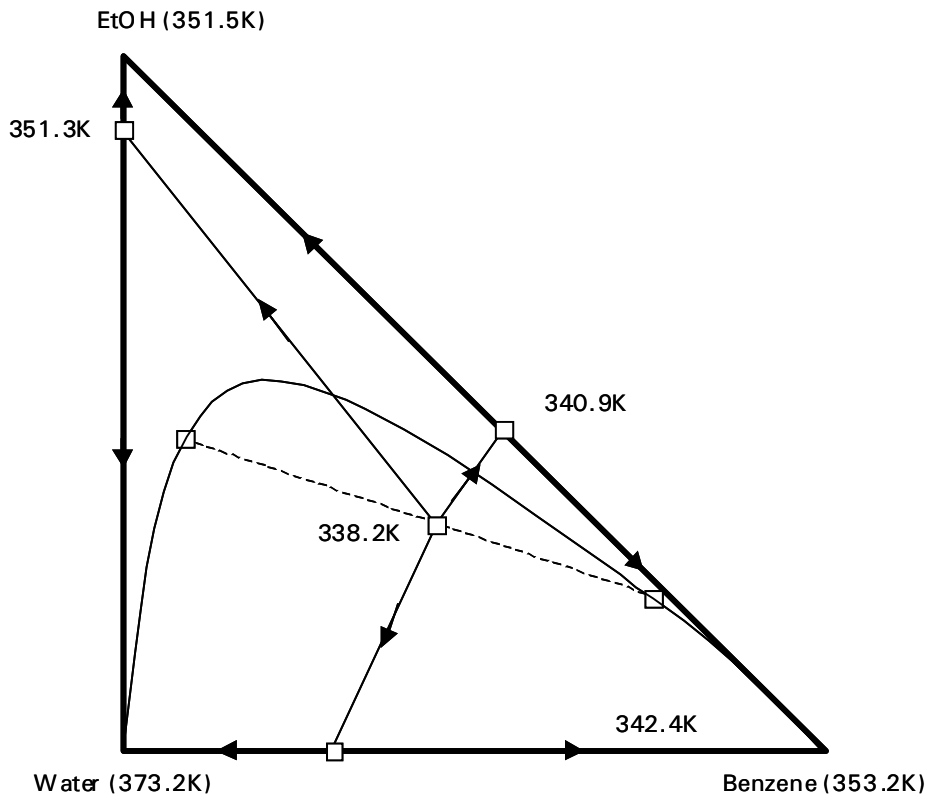


Figure 8: Residue Curve Map for Ethanol / Water / Benzene

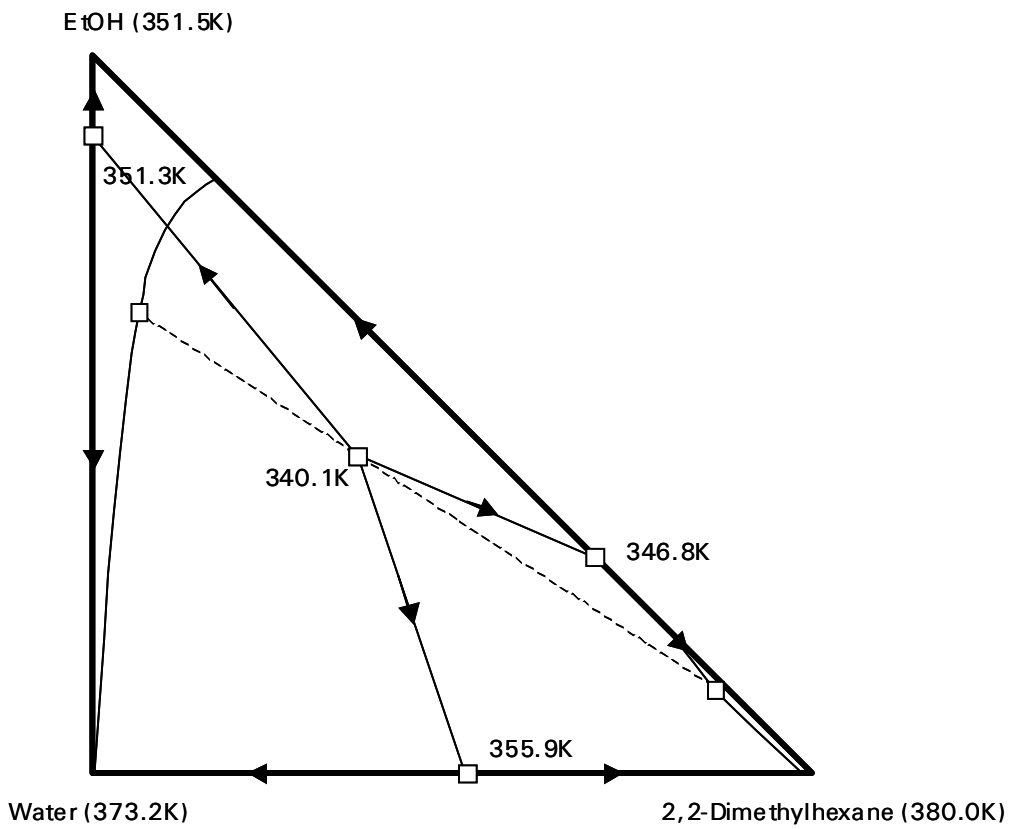


Figure 9: Residue Curve Map for Ethanol / Water / 2,2-Dimethylhexane

Water / Acetic Acid

The water acetic acid system may be separated by means of liquid-liquid extraction when the water concentration is above 50%. For low water concentrations, distillation may also be considered for this separation.

Ethyl acetate may be used as the entrainer in this separation process. It forms a binary heterogeneous azeotrope with water, as is shown in Table 11. The SolvGen algorithm was applied to this problem in order to find an alternative entrainer. The results are shown in Table 12.

Table 11: Industrial Entrainer for the Water / Acetic Acid System (350K)

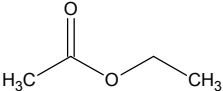

 Ethyl Acetate				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Water	0.3194	0.9913	0.1722	n/a
Acetic Acid	-	-	-	θ
Entrainer	0.6806	0.0087	0.8278	0.8208

Table 12: Entrainers for the Water / Acetic Acid System (350K)

 Tetrachloromethane (Carbon Tetrachloride)				
Component	Vapour	Liquid 1	Liquid 2	β_{12}
Water	0.3005	0.9998	0.0019	n/a
Acetic Acid	-	-	-	θ
Entrainer	0.6995	0.0002	0.9981	0.7008

Tetrachloromethane forms a binary heterogeneous azeotrope with water that is recovered in the distillate. As water and tetrachloromethane are almost completely immiscible, the two phases that form are almost pure. As acetic acid is not part of the azeotrope, the selectivity for acetic acid between the two liquid phases is not applicable in these systems.

The molar fractions of ethyl acetate and tetrachloromethane in their respective azeotropes do not differ significantly. The tetrachloromethane does however give a much wider separation between the liquid phase compositions than ethyl acetate.

For both entrainers the aqueous phase is almost pure water. With tetrachloromethane the organic phase is also almost pure, while a significant amount of water is present in the ethyl acetate organic phase. This water will be recycled to the column and will be continuously re-evaporated, increasing the energy use of the process.

Tetrachloromethane is toxic and this example is simply given to illustrate the power of SolvGen.

CONCLUSIONS

A Genetic Algorithm based CAMD method was presented for designing solvents for solvent driven separations. Extractive distillation and azeotropic distillation systems were illustrated, but liquid-liquid extraction and liquid chromatography have also been built into the SolvGen program. It was shown that solvent blends that perform better than pure solvents might be designed for extractive distillation systems. For heterogeneous azeotropic distillation it was shown that solvents that work better than the classical solvents may exist.

REFERENCES

1. K.G. Joback. and G. Stephanopoulos (1989)., Designing Molecules Possessing Desired Physical Property Values, FOCAPD'89, Snowmass CO, 363.
2. R. Gani and E.A. Brignole (1983), Fluid Phase Equilibria, 13, 331.
3. B. Nielsen, R. Gani and A. Fredenslund (1990), Computer-Aided molecular Design by Group Contribution Method." *Comp. App. Chem. Eng.*, 227.
4. R. Gani, B. Neilsen and A. Fredenslund (1991), *AIChE J.* 37(9): 1318.
5. E.J. Pretel, P.A. López, S.B. Bottini and E.A. Brignole (1994), *AIChE J.* 40(8): 1350.
6. V. Venkatasubramanian, K. Chan and J.M. Caruthers (1994) *Comp. Chem. Eng.* 18(9): 833.
7. C.D. Maranas (1996), *Ind. Eng. Chem. Res.*, 35, 3403.
8. N. Churi and L.E.K. Achenie, *Ind. Eng. Chem. Res.*, 35, 3788 (1996)
9. B. van Dyk (2001), Computerized Design of Solvents for Extractive Processes, PhD Thesis, University of Stellenbosch, South Africa.
10. B. van Dyk and I. Nieuwoudt (2000), *Ind. Eng. Chem. Res.* 39(5), 1423.
11. H.K. Hansen, P Rasmussen, A. Fredenslund, M. Schiller and J Gmehling (1991) *Ind. Eng. Chem. Res.* 30: 2352.
12. J. Gmehling, J Lohmann, A. Jakob, J. Li. and R. Joh (1998) *Ind. Eng. Chem. Res.* 37(12): 4876.
13. J. Marrero-Morejón and E. Pardillo-Fontdevila (1999) *AIChE J.* 45(3): 615.

14. K.G. Joback and R.C. Reid (1987) Chem. Eng. Comm. 57: 223.
15. M. Matsumoto and T. Nishimura (1998). "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudorandom Number Generator." ACM Trans. on Modeling and Computer Simulation 8(1): 3.
16. L.A. Robbins (1980) Chem. Eng. Prog. 76(10): 58.
17. J. Gmehling, J. Menke, J. Krafczyk and K. Fischer, K. (1994), *Azeotropic Data*, Vol. 1, VCH, Weinheim.