On the Design of Optimal Solvent Mixtures using Generalised Disjunctive Programming

Paul Akula, Polyxeni-M. Kleniati, Claire S. Adjiman

Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, London SW7 2AZ, United Kingdom

Abstract

We present a novel methodology for the formulation of the general mixture design problem which integrates Generalised Disjunctive Programming (Raman and Grossmann, 1994) into a Computer Aided Mixture Design framework. Two case studies are discussed: one aims at designing the most effective solvent mixture to dissolve ibuprofen and another at designing a solvent mixture for liquid-liquid extraction. In both cases, components of the optimal solvent mixture are selected from a list of up to nine molecules. The number of components, their identities and their mole fractions are optimised simultaneously to meet the specified performance criterion. The results of the case studies highlight the benefits of using mixtures.

Keywords: Optimal Solvent Mixture Design, Generalised Disjunctive Programming.

1. Introduction

The design of mixtures is an important and challenging problem with numerous industrial applications [1]. Of particular interest are applications in separation processes, such as liquid-liquid extraction, which require suitable solvents to meet given specifications [2]. Solvents are also used in chemical reactions to enhance the reaction rate [3] and in product design as constituents of the final product formulations [4].

However, the choice of solvents can have a significant impact on the performance of most processes as seen in crystallisation, where undesired crystal morphology can arise in some solvents, resulting in difficulties in the downstream processing and even in decreases in product performance for the case of drug manufacturing [5]. Furthermore, solvent mixtures are particularly desirable as they are known to achieve higher performance over pure solvents [6]. Hence, there is a need for a systematic approach to find optimal solvent mixtures.

Computer Aided Mixture Design (CAMbD, where b stands for “blend”) [7] has been used in various forms to determine the optimal solvent mixture. CAMbD methods generally use Mixed Integer Non-Linear Programming (MINLP) techniques to model the discrete decisions inherent in the solvent design problem [8-11]. However, in most existing formulations, the number of components in the mixture is predetermined and the identities of all but one of the mixture constituents are known a priori. Furthermore, modelling processes directly as MINLP problems can lead to severe numerical singularities for cases involving complex property models and a large combinatorial space [12].

In this paper, we formulate the optimal mixture design problem using Generalised Disjunctive Programming (GDP) [13]. GDP is a logic-based method, thus offering a
natural way to formulate the discrete choices in an optimisation problem. It has proved very successful in the design of complex processing networks [14,15] but has not yet been applied to molecular design. For a review of GDP techniques, the reader is referred to [16] and the references therein. We follow a systematic approach to developing GDP formulations for CAMBD, considering increasingly complex problems: we start with mixtures with a fixed number of components and progress to optimise both the number of components in the mixture as well as the components themselves. In particular, in solid-liquid and liquid-liquid equilibrium applications we select \( N \) solvents from a list, where \( N \) is either fixed, e.g. \( N \in \{1, \ldots, N_{\text{max}}\} \), or it can vary, i.e. \( 1 \leq N \leq N_{\text{max}} \). In the former case, disjunctions are required for each solvent molecule in the mixture, while in the latter case, disjunctions are required for the number of solvents and for the choice of solvents. The relevant solubility model equations are integrated within the disjunctions. Thus, numerical difficulties arising from the presence/absence of specific components can be avoided.

2. Problem Formulation

The proposed solvent mixture design GDP formulation optimises the number, choice and mole fractions of the solvents in the mixture for a specified performance measure subject to the appropriate constraints of the system. The basic steps for the formulation includes: defining disjunctions for the number of solvents, disjunctions for each choice of solvents, deriving logical conditions that express the relations between the disjunctive sets and finally posing the optimisation problem as an MINLP problem using the big-M relaxation approach in order to take advantage of the available algorithms for its solution.

The following notation is used throughout the paper: \( x \in \mathbb{R}^n \) is the vector of continuous variables expressing the mole fraction of each component and \( nc (nc_{\text{max}}) \) is the total number (maximum number) of components in the mixture. \( N (N_{\text{max}}) \) is the number (maximum number) of solvents in the mixture. Let also \( N' \) denote the number of non-solvent components in the mixture. It is clear that \( nc = N' + N \) and \( nc_{\text{max}} = N' + N_{\text{max}} \).

2.1. Disjunctions for the number of solvent molecules

The discrete choices for the number of solvents in the mixture are represented via disjunctions, spanning from 1 to \( N_{\text{max}} \). Each disjunction corresponds to the Boolean variables \( Y^N = \{\text{true, false}\}^{N_{\text{max}}} \), which control the feasible space in which the continuous variables \( x \) lie. Relevant equations that depend on the number of components in the mixture are placed in the appropriate disjunction:

\[
\bigvee_{N=1, \ldots, N_{\text{max}}} \begin{bmatrix}
Y^N \\
\mathbf{f}_{N+1} (x) = 0 \\
x_j \geq 0.001, i = N'+1, \ldots, N' + N \\
x_j = 0, i = N' + N, \ldots, nc_{\text{max}}
\end{bmatrix}.
\]  

Constraints \( \mathbf{f}_{N+1} (x) \) express property constraints, such as phase equilibrium constraints, and the formulation above ensures that these only hold true when the corresponding Boolean variable \( Y^N \) is true. A minimum mole fraction of e.g. 0.001 is imposed for any solvent component which is chosen to appear in the mixture.

2.2. Disjunctions for the choice of each solvent molecule

Boolean variables \( Y_{i,s} \), \( i = N'+1, \ldots, nc \) and \( s = 1, \ldots, ns \), are defined for each solvent that establishes whether or not a solvent molecule is selected in the mixture. The constraints
On the design of optimal solvent mixtures using GDP

$h_{i,s}$ ensure that the properties/parameters describing the chosen component are correctly assigned to solvent $s$. The corresponding disjunctions are:

$$\forall_{s=1,...,ns} \left[ Y_{i,s} \right]_{h_{i,s}} = 0, i = N'+1,...,nc_c, \tag{D2}$$

where $ns$ is the number of solvents in the list of solvents.

2.3. Logic propositions

To finalise the proposed formulation, we also need to define appropriate logical relations in order to ensure that only one disjunction is chosen:

$$\sum_{n=1}^{ns} Y_n = 1, \tag{3}$$

at most $N$ solvents are selected:

$$\sum_{i=1}^{nc} Y_{i} \leq 1 \forall i = N'+1,...,nc_{max}, \tag{4}$$

and the selected solvents are ordered in order to avoid degeneracy:

$$Y_{i} \Rightarrow \neg Y_{i', s}, i = N'+1,...,nc-1, s = 2,...,N, i' = i+1,...,nc, s' = 1,...,s. \tag{5}$$

Additional logical relations are also needed to ensure that the Boolean variables $Y_n$ control the Boolean variables $Y_{i}$ appropriately [17]. In what follows, let $\Omega(Y)$ express logical relations (3) - (5).

2.4. GDP formulation

The proposed GDP formulation is as follows:

$$\min/\max P_{mix}$$

s.t. $g(x) \leq 0$,

$$(D1), (D2), \Omega(Y) = true,$$

$$x \in \mathbb{R}^{nc}, Y = (Y_n, Y_{i,s}) \in \{true, false\}^{ns \times (nc_{max})}, \tag{6}$$

where $P_{mix}$ is a performance objective optimised subject to a set of constraints $g(x)$, e.g. physical properties, that must hold regardless of the discrete choices.

3. Examples

Two case studies are discussed here. All problem instances were solved in GAMS/DICOPT 21.3 and tested on a 64-bit Intel Xeon running at 1.60GHz.

3.1. Extraction of Acetic Acid from Water

This example is similar to the problem found in [18] and involves finding the solvent or solvent mixture that requires a minimum solvent-to-feed ratio for extraction of acetic acid/water mixture of 8 wt% acetic acid and feed flow rate of 13500 kg hr$^{-1}$ at 298 K. The raffinate should contain at most 0.3 mol% of acetic acid. We present the numerical results for three problem instances for a fixed number of solvents, i.e. for $N=1,2,3$. Fixing $N$ in this case study corresponds to fixing $Y_n$ in the disjunctions (D1). Hence, only disjunctions (D2) need to be defined and the appropriate logical relations. The objective function involves minimising the sum of the solvent-to-feed ratio and the composition of the water ($W$) in the extract phase minus the composition of acetic acid (AA) in the extract phase [19]. The mole balances and the phase equilibrium constraints are included in the model, while the modified UNIFAC (Dortmund) equation is used to predict the activity coefficients. The three problem instances and the design space are
summarised in Table 1. The mixture of hexanone and pentanol gave the best objective value of 2.229 and a recovery of 94.4 %, as shown in Table 2. This indicates that solvent mixtures can outperform single solvents.

Table 1: Problem specification for the extraction of acetic acid from water

<table>
<thead>
<tr>
<th>All components</th>
<th>N=1</th>
<th>N=2</th>
<th>N=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>{AA,W,S1}</td>
<td>[AA,W,S1,S2]</td>
<td>[AA,W,S1,S2,S3]</td>
<td></td>
</tr>
</tbody>
</table>

| Solvent mixture | [S1]               | [S1,S2]            | [S1,S2,S3]         |

| List of solvents | {hexanone, heptanol, pentanol, butyl acetate, isopropyl acetate, di-methyl heptane, tetra-methyl hexane} |

| Atom groups      | {CH₃, CH₂, CH, CH=C, OH(p), OH(s), H₂O, CH₃CO, CH₃COO, COOH, CyCH₂, CyCH} |

Table 2: Results summary for extraction of acetic acid from water

<table>
<thead>
<tr>
<th>Optimal Obj. Value</th>
<th>Solvent mixture</th>
<th>Composition (mol/mol)</th>
<th>Recovery %</th>
<th>No. binary vars</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N=1 2.320</td>
<td>[hexanone]</td>
<td>[1]</td>
<td>92.2</td>
<td>8</td>
<td>0.86</td>
</tr>
<tr>
<td>N=2 2.229</td>
<td>[hexanone, pentanol]</td>
<td>[0.834, 0.166]</td>
<td>94.4</td>
<td>16</td>
<td>42.25</td>
</tr>
<tr>
<td>N=3 2.240</td>
<td>[hexanone, heptanol, pentanol]</td>
<td>[0.833, 0.10, 0.157]</td>
<td>94.3</td>
<td>24</td>
<td>482.97</td>
</tr>
</tbody>
</table>

3.2. Solubility of Ibuprofen

In this example, instances with fixed N were solved, N=1,2,3, and one instance with N as unknown, i.e. 1 ≤ N ≤ 3. In all instances, the optimal mixture of N solvent(s) that maximises the solubility of Ibuprofen (Ibu) at 300 K was sought. The UNIFAC model is employed. The problem instances and the design space are in Table 3. Disjunctions (D1) for the number of solvents and disjunctions (D2) for the choice of each solvent were defined. The best performance is found for mixtures of two or three components, again showing that solvent mixtures achieve higher performance than pure solvents.

Table 3: Problem specification for ibuprofen solubility case study

<table>
<thead>
<tr>
<th>1 ≤ N ≤ 3</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>All components</td>
<td>{Ibu,S1,S2,S3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solvent mixture</td>
<td>[S1,S2,S3]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>List of solvents</td>
<td>{chloroform, methanol, ethanol, acetone, MIBK, ethyl acetate, 2-propanone, toluene, water}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atom groups</td>
<td>{CH₃, CH₂, CH, aCH, aCCH₂, aCCH₃, aCCH, OH, CH₃OH, H₂O, CH₃CO, CH₃COO, COOH, CHCl₃}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Results summary for ibuprofen solubility case study

<table>
<thead>
<tr>
<th>N</th>
<th>x_{ib}</th>
<th>S1</th>
<th>x_{s1}</th>
<th>S2</th>
<th>x_{s2}</th>
<th>S3</th>
<th>x_{s3}</th>
<th>No. binary vars</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3180</td>
<td>CHCl_3</td>
<td>0.6820</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>0.3338</td>
<td>CHCl_3</td>
<td>0.5229</td>
<td>CH_3OH</td>
<td>0.1432</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>0.3311</td>
<td>CHCl_3</td>
<td>0.5200</td>
<td>CH_3OH</td>
<td>0.1369</td>
<td>C_2H_5OH</td>
<td>0.01</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>≤ 3</td>
<td>0.3338</td>
<td>CHCl_3</td>
<td>0.5229</td>
<td>CH_3OH</td>
<td>0.1432</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

*value at lower bound

4. Conclusions

A new methodology of formulating the mixture design problem which optimises the number of components, their identities and compositions has been presented. GDP is integrated into the CAM^D framework for efficient modeling which tackles the discrete choices inherent in mixture design and avoids numerical difficulties associated with the absence of components. This promising framework is currently being extended to designing molecules from groups and its computational performance is being investigated.

References