Vaccine Manufacturing Modelling Webinar

Developing Countries Vaccine Manufacturers Network (DCVMN)

13 December 2018

Zoltan Kis, Maria Papathanasiou, Cleo Kontoravdi and Nilay Shah
Centre for Process Systems Engineering
Chemical Engineering
• Vaccine Manufacturing Hub

• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (manufacturing and supply chain level).
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (supply chain level).
WS1. Integrating life science and engineering to improve existing platforms

- LMIC partners have platforms for the production of vaccine formulations suitable for LMIC environment
- WS1 focuses on how best to exploit and further develop these to
  - enable mass vaccination campaigns
  - deal effectively with new outbreaks
- Objectives:
  - operational efficiency for cost reduction
  - rapid response of existing assets
  - end-to-end system design
- How? Whole process analysis and optimisation to address bottlenecks
  - In Life Sciences: host cell system or vector optimisation for improved productivity/quality
  - In Engineering: downstream separations, formulation and packaging
Current capabilities

- 90% of the LMIC vaccine production is attenuated and inactivated bacterial/virus vaccines
- 10% mostly conjugates

We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs

- Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives

Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency

Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing
WS1. Methodology

- Model, simulate and optimize the manufacturing and delivery processes at:
  1. unit operation level using gPROMS;
  2. process flow sheet level using SuperPro Designer,
  3. supply chain level using GAMS
- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.
- Enhance bioprocessing by:
  a) process intensification continuous processing strategies;
  b) de-bottlenecking;
  c) “process telescoping” combining several unit operations into one (e.g. charge and size based separation in one unit operation).
- Integration with experimental work, iterative
Advantages of a modular design:

- Plug & Play
- Standardization
- Flexibility
- Streamlined troubleshooting
- Easy of training operators
WS1. Rapid prototyping of novel downstream separation process concepts

- Explore two purification concepts for whole virus/bacteria, sub-unit and proteins:
  - “process telescoping” whereby several unit operations are combined into one (e.g. expanded bed affinity adsorption combining solids removal, capture and primary purification)
  - continuous operation (e.g. moving to continuous chromatography using simulated moving bed technology).
- Our key activities will involve high throughput experiments, models and big data analytics.
- Deliverable: Demonstration of new vaccine separation design concepts at lab scale
WS1. Key drivers for continuous manufacturing

- Manufacturing capacity
- Reduced footprint, labour costs and CapEx
- Flexibility
- Speed to market
- Improved quality through the application of QbD & PAT
# WS1. Challenges

## Design
- Design and control interactions
- Bioreactor design
- Downstream setup configuration

## Control
- Complex operation profiles
- Complicated process models
- Unavailable measurements

## Operation
- Optimal setup configuration
- Feasible operation
- Task coordination (scheduling)

## Regulatory Bodies
- Enhanced process understanding
- Monitoring
- Global control
WS1. Requirements for a smooth transition

- Quality by Design (QbD)
- Thorough understanding
- Identification of risks/bottlenecks
- Process monitoring
- Global process control

… minimizing experimental time & cost….
Scaled up designs will be used to explore supply chain configuration:
- centralised vs decentralised
- shipment of bulk or fully-filled vaccines to clinics/local fill-finish plants

What is a Supply Chain?
- The alignment of firms that bring products or services to market
- Linked by counter-current flow of material and information

Supply Chain Management
“The systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole.”
WS1. Results

- State of the art conventional bacterial, viral and recombinant vaccine manufacturing processes have been reviewed.

- Calculated the batch volume for producing 1500 doses of the Hand, foot, and mouth disease (HFMD) vaccine in yeast at Dalian Hissen BioPharm Co., Ltd. (China).
WS1. Ongoing case studies

- Analysing and improving vaccine manufacturing processes at MSD Wellcome TrustHilileman Laboratories Pvt. Ltd. (Hilleman Labs, India) for Recombinant cholera toxin B subunit and whole cell cholera vaccine production.

- Developing a model of the Vero cell based whole viral vaccine production and perform sensitivity analysis to quantify the relationships between CPPs and CQAs in support of QbD of this family of vaccine manufacturing processes. In collaboration with VaBiotech (Vietnam).

- We are engaging with additional vaccine developers and manufacturers in developing countries to simulate and optimize their processes (e.g. DCVMN meeting and webinars).
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (supply chain level).
**WS2. The 4 emerging vaccine platform technologies**

**RNA vaccines**
- ~10 kb self-amplifying RNA
- regular mRNA
- produced by *in vitro* transcription

**ADDomer vaccines**
- adenovirus dodecahedron derived multimer
- 3.6 MDa VLP, 360 configurable epitopes, different epitope types
- insect cell-baculovirus expression

**Humanized yeast vaccines**
- glycoengineered yeast
- high-yield yeast expression of recombinant proteins

**GMMA vaccines**
- Generalized Modules for Membrane Antigen, outer membrane vesicles, 20-60 nm diameters
- configurable membrane proteins, bacterial expression
## WS2. Platform evaluation and improvement metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speed</strong></td>
<td>~100,000 vaccine doses, weeks after threat antigen identification</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>low cost, below 1 $/dose</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>on-demand production of a wide range of vaccine types (viral and bacterial)</td>
</tr>
<tr>
<td><strong>Technological complexity</strong></td>
<td>low technological complexity for implementation in developing countries</td>
</tr>
<tr>
<td><strong>Technology readiness</strong></td>
<td>mature technologies with licensed products &amp; established manufacturing processes</td>
</tr>
<tr>
<td><strong>Ease of scale-up or -out</strong></td>
<td>highly scalable upstream and downstream processes</td>
</tr>
<tr>
<td><strong>Thermo-stability of product</strong></td>
<td>vaccines stable at 40°C for at least 6 months</td>
</tr>
</tbody>
</table>
WS2. Workflow

- **Compare**
  - Assess and compare the techno-economic feasibility of the 4 emerging vaccine technologies

- **Optimize**
  - Optimize unit operations and process flows

- **Detail Model**
  - For the techno-economically most feasible platform build a detailed and optimized process model

- **Pilot**
  - In the next phase of the project, based on the optimized process model, build a demonstration pilot plant in Uganda, at clinical trial scale.

- **Production**
  - After clinical trials or during epidemics, scale production up or out to meet demand.
WS2. Methodology

- To overcome data scarcity and uncertainty:
  1. Calculate parameters on a first-principle, bottom-up basis;
  2. Estimate parameters based on similar processes and products by surveying the scientific literature and patent databases, and by interviewing experts;
  3. Model, simulate and optimize the manufacturing processes at unit operation level (using gPROMS) and process flow sheet level (using SuperPro Designer, Aspen Batch Process Developer), at supply chain level (using GAMS)

- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.

- Enhance bioprocessing by: (a) evaluating process intensification continuous processing strategies; (b) de-bottlenecking; (c) “process telescoping” combining several unit operations into one
WS2. RNA platform results

- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5’ capping, ARCA

Cost per dose at 20 µg/dose: 0.72 USD/dose
WS2. ADDomer platform results

- **adenovirus dodecahedron derived multimer**
- **3.6 MDa VLP, 360 configurable epitopes, different epitope types**
- **insect cell-baculovirus expression**
- **intracellular**
WS2. Humanized yeast platform results

- glycoengineered yeast
- high-yield yeast expression of recombinant proteins
- extracellular
• Generalized Modules for Membrane Antigen, outer membrane vesicles, 20-60 nm diameters
• configurable membrane proteins, bacterial expression
• extracellular
## Table 1. Feasibility and risk assessment of the 4 emerging platform technologies

<table>
<thead>
<tr>
<th>Metric</th>
<th>Platform a)</th>
<th>Yeast platform</th>
<th>ADDomer platform</th>
<th>GMMA platform</th>
<th>RNA platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Technology readiness</td>
<td></td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>2 Technological complexity</td>
<td></td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3 Ease of scale-up and -out</td>
<td></td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4 Flexibility b)</td>
<td></td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5 Thermo-stability of product</td>
<td></td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6 Speed of response</td>
<td></td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td><strong>Sum: overall feasibility and risk estimate c)</strong></td>
<td>16</td>
<td>18</td>
<td>22</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

a) Yeast platform - Humanised, high-yield yeast platform for recombinant vaccine manufacturing; ADDomer platform - Insect cell-baculovirus platform for recombinant vaccine manufacturing; GMMA platform - Outer membrane vesicle vaccines, Generalized Modules for Membrane Antigen vaccine manufacturing; RNA platform - RNA vaccine manufacturing.

b) Universal applicability for the manufacturing of a wide range of vaccines.

c) The overall feasibility and risk estimate was calculated by summing up the values for each metric per technology.
Process Systems Engineering – lifecycle view
Key outcomes from this workstream include:

- A screening methodology which can identify which platform is best suited to a particular vaccine
- Conceptual and detailed demonstration industrial process designs/blueprints which build on lab data
- Actual physical demonstrations of new manufacturing concepts with emerging data
- Application of these to demonstrate the benefits of novel platforms and approaches as an evidence base for regulators, healthcare providers and manufacturers.
WS2. Future tasks

Task 2.3: Application of detailed demonstration design methodology to up to 6 shortlisted demonstration process concepts, leading to:
   (a) detailed designs and
   (b) more accurate KPIs including cost,
which will be used to select the demonstration projects for physical deployment.

Task 2.4: Development of physical demonstrators - up to 6 scale appropriate physical demonstrators used to:
   • advance the knowledge around industrialisation of the new platforms,
   • evaluate actual performance in an industrial setting,
   • feed back information to the model-based design activity to support analysis and design of full commercial scale processes.
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (supply chain level).
Methodology

- We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs
  - Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives
- Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency
- Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing
Why model?

Competitive advantage

- Innovation

Impediments

- Higher risks
- Effective risk management
- Limited scope for evaluation of alternatives

Imperatives

- Speed (time-to-market)

Modelling

- New equipment & process designs
- New chemistry & catalysts
- New materials of construction

(Costas Pantelides, PSE Ltd)
Innovation, risk and modelling – models and data not models or data

Competitive advantage

Imperatives

- Innovation
- Speed (time-to-market)

Impediments

- Higher risks
- Effective risk management
- Limited scope for evaluation of alternatives

Integrated experimental & modelling methodologies

Modelling
Employ high-level models of traditional upstream systems (bacteria, yeast and animal cells) and our emerging platform

Optimise using process mapping, bottleneck identification and process intensification, building on work in biologics manufacturing.

Also identify raw materials and suitable alternatives available locally and/or at lower cost.

Deliverable: Demonstration of benefits of integrated approach on primary production systems
Development of computational models for whole systems analysis

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation
- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study
Modelling for process optimisation

Following Quality by Design principle for increasing process understanding
Development of reliable process models

Cell Culture Dynamics

Intracellular kinetic model
Cell Culture Dynamics

In silico optimisation

Intracellular kinetic model

Golgi model

Cell culture time (days)

Visible cell density (Billion cells/L)

Galactosylation

Control - model result

Control

Optimised model output

Control - model result

Control

Optimised model output
Experimental results of optimal operating conditions
Whole process analysis: intracellular product
Initial process design and dynamic optimisation

Figure 3. Optimal fermenter profiles, for problem 1 (Table 1). --- P; ---- S_i; --- X_i; --- μ.

Figure 5. Stress and cell strength distribution in homogenizer, problem 1. ---- D_i; ■ f-D_i; ● f_s.
Simultaneous design of key variables

Table 2. Optimal design for batch fermentation, maximizing profit, purity ≥ 0.13.

<table>
<thead>
<tr>
<th></th>
<th>Fermenter</th>
<th>Cell harvesting centrifuge</th>
<th>Homogenizer</th>
<th>Centrifuge 1</th>
<th>Precipitation stage 1</th>
<th>Centrifuge 2</th>
<th>Precipitation stage 2</th>
<th>Centrifuge 3</th>
<th>Overall cycle time</th>
<th>Process performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( S_p )</td>
<td>( X_0 )</td>
<td>( V (= V_0) )</td>
<td>( N )</td>
<td>( t_{f\text{om}} )</td>
<td>( F_{\text{red}} )</td>
<td>( n_{\text{om}} )</td>
<td>( t_{\text{cnt1}} )</td>
<td>( M_{\text{at.m}} )</td>
<td>( M_{\text{at.s}} )</td>
</tr>
<tr>
<td>Fermenter</td>
<td>101.04</td>
<td>5</td>
<td>800</td>
<td>1</td>
<td>11.888</td>
<td>110</td>
<td>7.273</td>
<td>10</td>
<td>7224.3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell harvesting centrifuge</td>
<td>( F_{\text{red}} )</td>
<td>110</td>
<td>( n_{\text{om}} )</td>
<td>7.273</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homogenizer</td>
<td>( F )</td>
<td>60</td>
<td>( n_{\text{om}} )</td>
<td>13.333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 1</td>
<td>( F_{\text{red}} )</td>
<td>80</td>
<td>( n_{\text{om}} )</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precipitation stage 1</td>
<td>( M_{\text{at.m}} )</td>
<td>0</td>
<td>( n_{\text{om}} )</td>
<td>32000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 2</td>
<td>( F_{\text{red}} )</td>
<td>80</td>
<td>( n_{\text{om}} )</td>
<td>6500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precipitation stage 2</td>
<td>( M_{\text{at.m}} )</td>
<td>32000</td>
<td>( n_{\text{om}} )</td>
<td>28000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 3</td>
<td>( F_{\text{red}} )</td>
<td>75</td>
<td>( n_{\text{om}} )</td>
<td>9500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall cycle time</td>
<td>( t_{\text{pc}} )</td>
<td>13.333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process performance</td>
<td>profit</td>
<td>48311</td>
<td>purity</td>
<td>0.1299</td>
<td>recovery</td>
<td>0.7585</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Optimal design for batch fermentation, maximizing purity.

<table>
<thead>
<tr>
<th></th>
<th>Fermenter</th>
<th>Cell harvesting centrifuge</th>
<th>Homogenizer</th>
<th>Centrifuge 1</th>
<th>Precipitation stage 1</th>
<th>Centrifuge 2</th>
<th>Precipitation stage 2</th>
<th>Centrifuge 3</th>
<th>Overall cycle time</th>
<th>Process performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( S_p )</td>
<td>( X_0 )</td>
<td>( V (= V_0) )</td>
<td>( N )</td>
<td>( t_{f\text{om}} )</td>
<td>( F_{\text{red}} )</td>
<td>( n_{\text{om}} )</td>
<td>( t_{\text{cnt1}} )</td>
<td>( M_{\text{at.m}} )</td>
<td>( M_{\text{at.s}} )</td>
</tr>
<tr>
<td>Fermenter</td>
<td>107.61</td>
<td>5</td>
<td>800</td>
<td>1</td>
<td>11.655</td>
<td>115.85</td>
<td>9042.3</td>
<td>6.905</td>
<td>13.333</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell harvesting centrifuge</td>
<td>( F_{\text{red}} )</td>
<td>115.85</td>
<td>( n_{\text{om}} )</td>
<td>9042.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homogenizer</td>
<td>( F )</td>
<td>60</td>
<td>( n_{\text{om}} )</td>
<td>13.333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 1</td>
<td>( F_{\text{red}} )</td>
<td>60</td>
<td>( n_{\text{om}} )</td>
<td>2274.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precipitation stage 1</td>
<td>( M_{\text{at.m}} )</td>
<td>0</td>
<td>( n_{\text{om}} )</td>
<td>31114</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 2</td>
<td>( F_{\text{red}} )</td>
<td>70.376</td>
<td>( n_{\text{om}} )</td>
<td>7391.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precipitation stage 2</td>
<td>( M_{\text{at.m}} )</td>
<td>31114</td>
<td>( n_{\text{om}} )</td>
<td>18945</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centrifuge 3</td>
<td>( F_{\text{red}} )</td>
<td>60</td>
<td>( n_{\text{om}} )</td>
<td>8000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall cycle time</td>
<td>( t_{\text{pc}} )</td>
<td>15.946</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process performance</td>
<td>profit</td>
<td>12505</td>
<td>purity</td>
<td>0.1395</td>
<td>recovery</td>
<td>0.7191</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From design to operations

Table 1. Optimal storage tank capacities.

<table>
<thead>
<tr>
<th>State</th>
<th>Tank capacity (l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>feed and product states</td>
<td>not sized</td>
</tr>
<tr>
<td>broth</td>
<td>800</td>
</tr>
<tr>
<td>cells</td>
<td>24.48</td>
</tr>
<tr>
<td>dcells</td>
<td>740</td>
</tr>
<tr>
<td>int1</td>
<td>125</td>
</tr>
<tr>
<td>liquid1</td>
<td>684</td>
</tr>
<tr>
<td>ppt1</td>
<td>760</td>
</tr>
<tr>
<td>liquid2</td>
<td>684</td>
</tr>
<tr>
<td>ppt2</td>
<td>652.34</td>
</tr>
</tbody>
</table>

Figure 1. State task network for the example problem.

Figure 4. Cyclic schedule for one batch, as above, showing three batches.
Debottlenecking of influenza vaccine process
Debottlenecking model
Debottlenecking

Table 2. Comparisons of different methods debottlenecking strategies.

<table>
<thead>
<tr>
<th>Case</th>
<th>Butches obtained (% change)</th>
<th>Bottleneck</th>
<th>Customer service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>67</td>
<td>Incubation</td>
<td>93%</td>
</tr>
<tr>
<td>B</td>
<td>93 (39%)</td>
<td>Incubation</td>
<td>100%</td>
</tr>
<tr>
<td>C</td>
<td>98 (46%)</td>
<td>Inactivation</td>
<td>100%</td>
</tr>
<tr>
<td>D</td>
<td>98 (46%)</td>
<td>Incubation</td>
<td>100%</td>
</tr>
</tbody>
</table>

with cleaning period of 8 hours per night
Incorporation of uncertainty

Figure 1: The fermentor tank

Figure 2: Expected profit vs $V_{max}$

Figure 3: Profit distribution for nominal case

Figure 4: Product concentration distribution for nominal case

Figure 5: Profit distribution for robust case

Figure 6: Product concentration distribution for robust case
The aid of computational tools

- **Design**
  - Re-design current state-of-the-art
  - Study dynamics
  - Thorough system understanding

- **Experiments**
  - Only the necessary
  - Minimize labor time & cost

- **Modeling**
  - Dynamic
  - High-fidelity
  - Testing platform

- **Control**
  - Monitoring
  - Optimal performance guaranteed

- **Optimization**
  - Optimal operating conditions
  - Optimal performance

The aid of computational tools aims to:

- Re-design current state-of-the-art
- Study dynamics
- Thorough system understanding
- Only the necessary
- Minimize labor time & cost
- Dynamic
- High-fidelity
- Testing platform
- Monitoring
- Optimal performance guaranteed
- Optimal operating conditions
- Optimal performance
Development of computational models for whole systems analysis

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation

- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (supply chain level).
Introduction to gPROMS

Describing new processes and unit operations

Dynamic process modelling
Seamless modelling capabilities (no manual discretization required)
Other functionalities: dynamic optimisation, parameter estimation based on experimental data, experiments to be designed
Main navigation window

Key entities and functionalities
Declaration of variables & bounds
**Declaration of:**
- Parameters
- Variables
- Equations

<table>
<thead>
<tr>
<th>Parameter/Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no_species</td>
<td>A3 INTEGER</td>
<td>Number of Species</td>
</tr>
<tr>
<td>Vr</td>
<td>A3 REAL</td>
<td>CSTR Volume</td>
</tr>
<tr>
<td>alpha</td>
<td>A3 REAL</td>
<td>Protein Synthesis Coefficient</td>
</tr>
<tr>
<td>beta</td>
<td>A3 REAL</td>
<td>Fatty Acid Synthesis Coefficient</td>
</tr>
<tr>
<td>gamma</td>
<td>A3 REAL</td>
<td>Fatty Acid Mobilization Coefficient</td>
</tr>
<tr>
<td>mu_bar</td>
<td>A3 REAL</td>
<td>Theoretical Maximum Growth Rate</td>
</tr>
<tr>
<td>Qo</td>
<td>A3 REAL</td>
<td>Minimal Nitrogen Quota</td>
</tr>
<tr>
<td>cm</td>
<td>A3 REAL</td>
<td>Maximal Uptake Rate</td>
</tr>
<tr>
<td>Rs</td>
<td>A3 REAL</td>
<td>Half Saturation Constant</td>
</tr>
<tr>
<td>F</td>
<td>A3 flow_rate</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>Frec</td>
<td>A3 flow_rate</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>Fr</td>
<td>A3 flow_rate</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>rset</td>
<td>A3 flow_rate</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>Fout</td>
<td>A3 flow_rate</td>
<td>Flow Rate</td>
</tr>
<tr>
<td>s</td>
<td>A3 concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>s_i</td>
<td>A3 concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>s_r</td>
<td>A3 concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>rec</td>
<td>A3 ARRAY(no_species) OR concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>r</td>
<td>A3 ARRAY(no_species) OR concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>ex</td>
<td>A3 ARRAY(no_species) OR concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>q</td>
<td>A3 concentration</td>
<td>Concentration</td>
</tr>
<tr>
<td>qn</td>
<td>A3 no_type</td>
<td>No Type</td>
</tr>
<tr>
<td>q1</td>
<td>A3 no_type</td>
<td>No Type</td>
</tr>
<tr>
<td>qf</td>
<td>A3 no_type</td>
<td>No Type</td>
</tr>
<tr>
<td>miu</td>
<td>A3 rate</td>
<td>Rate</td>
</tr>
<tr>
<td>c</td>
<td>A3 rate</td>
<td>Rate</td>
</tr>
</tbody>
</table>
Model entity

- Description of equations
- Possible to include:
  - ODEs
  - DAEs
  - PDAEs
• Link process to model
• Assign values to:
  - Parameters
  - Variables
Parameter estimation

- Performed experiments
- Experimental data
- Experimental error
Parameter estimation

- Simulation vs experimental data
- Here: estimation of 1 parameter
- Report generated on parameter accuracy
Open loop model simulation
• Declaration of process profile
• Declaration of control variables (manipulated)
• Declaration of constraints
Dynamic optimisation – Results

- Suggested optimal values following 5 iterations
Getting Help

- In gPROMS click on “Help” menu then click on “Documentation”.

- PSE Webinars:
  https://www.psenterprise.com/events/webinars
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level);
  • SuperPro Designer (process flow sheet level);
  • GAMS (supply chain level).
Introduction to SuperPro Designer

- Process flow sheet level modelling;
- Facilitates modelling, evaluation and optimization;
- Graphical user interface;
- Works by solving material and energy balances;
- Can be interfaced with Visual Basic for Applications (VBA), MatLab, Python, for automation and sensitivity analysis.
SuperPro Designer – Key features

- Contains models for over 140 unit procedures/operations;
- Extensive chemical component and mixture database;
- Extensive equipment and resource databases;
- Equipment sizing and costing;
- Thorough process economics;
- Scheduling of batch operations;
- Throughput Analysis and Debottlenecking.
SuperPro Designer – Typical questions being addressed

- How would the cost per dose change with increasing titres/yields?
- Which process design is more cost effective (by comparing different process designs)?
- How would the cost per dose and the upfront capital cost change at different production scales?
- Where are the production bottlenecks?
- Which are the major cost components and how to reduce those?
- How feasible is a continuous process compared to a batch process?
SuperPro Designer – Graphical User Interface (GUI)

- Material, energy, labour inputs and process parameters are defined using GUI
SuperPro Designer – Model inputs

- Type of unit procedures and parameters for each operation within procedures;

- Material inputs (default values are provided for labour and utilities);

- Definition of the reaction/fermentation (e.g. stoichiometric, kinetic, equilibrium) and reaction components;

- Specification of material flows between unit procedures;

- Specification of the sequence and duration of unit procedures (i.e. scheduling);

- Optional: a variety of other inputs can be specified (e.g. costs of equipment, costs of consumables, costs of labour and utilities, etc.)
SuperPro Designer – Outputs

Charts:
- Equipment occupancy charts
- Operations Gantt Charts
- Equipment Gantt Charts
- Material, labour and utility utilization charts

Reports (MS Word, MS Excel or PDF formats):
- Materials & Streams
- Economic Evaluation
- Cash Flow Analysis
- Itemized Cost
- Environmental Impact, etc.
SuperPro Designer
Example Problem 1
“Cost modelling for new vaccine processes”
RNA vaccines – how expensive? how to reduce costs?

- ~10 kb self-amplifying RNA
- produced by in vitro transcription
- cell-free product
- Co-transcriptional 5’ capping, ARCA
RNA vaccines – how expensive? how to reduce costs?

- ~10 kb self-amplifying RNA
- produced by in vitro transcription
- Co-transcriptional 5’ capping, ARCA
- 5x Recycling of materials
RNA production model in SuperPro Designer

- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5’ capping, ARCA

Cost per dose at 20 µg/dose at 1 L reaction volume: 0.72 USD/dose *

* Costs do not include formulation and secondary manufacturing (fill and finish) costs, upstream costs were accounted for indirectly. Default capital, labour, consumables, utilities, facility-related, maintenance and QA/QC costing values and methods from SuperPro Designer were used.
RNA Cost modelling results

* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly;
Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.
RNA Cost modelling results

- Capital costs
- Material costs
- Labour costs
- Facility-Dependent costs
- Other (QC/QA, cons., util., etc.)

* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly; Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.
RNA Cost modelling results

<table>
<thead>
<tr>
<th>Recirculation?</th>
<th>No</th>
<th>5x</th>
<th>5x</th>
</tr>
</thead>
<tbody>
<tr>
<td>5’ Cap type?</td>
<td>ARCA</td>
<td>ARCA</td>
<td>Non-capped</td>
</tr>
</tbody>
</table>

- **Cost per dose [USD/dose]**
  - Polycationic formulation
  - Fill to Finish
  - RNA bulk cost at 20 μg/dose
• Pressing the F1 button;


• Training videos, from the Intelligen, Inc. website: http://www.intelligen.com/videos.html

• Papers and literature, from the Intelligen, Inc. website: http://www.intelligen.com/literature.html

• Attending training course offered by Intelligen, Inc.: http://www.intelligen.com/training.html
• Work Stream 1 (WS1): Improving existing processes and platforms

• Work Stream 2 (WS2): Design, development and implementation of new platform technologies

• Modelling tools:
  • gPROMS (unit operation level); ✓
  • SuperPro Designer (process flow sheet level); ✓
  • GAMS (manufacturing and supply chain level).
Introduction to GAMS

GAMS Example Problem

Exercise

Latest GAMS version (demo) free download link:
http://www.gams.com/download/

General Algebraic Modelling System

is an

Algebraic Modelling Language (AML)
(high-level computer programming language)

main advantages

The syntax is similar to the mathematical notation of optimization problems.
The programming of the optimization model is independent of the solution algorithm.
GAMS is a modeling system for optimization that provides an interface with a variety of optimization algorithms (solvers).

Users introduce the model to GAMS in the form of algebraic equations. GAMS compiles the model and interfaces automatically with a solver.
GAMS user interface
GAMS optimization model types and solvers

Main optimization model types in GAMS:

- **LP** Linear Programming
- **NLP** NonLinear Programming
- **MIP** Mixed Integer linear Programming
- **RMIP** relaxed **MILP** where the integer variables are treated as continuous
- **MINLP** Mixed Integer NonLinear Programming; involve integer variables and nonlinear equations
- **RMINLP** Relaxed **MINLP** where the integer variables are treated as continuous

Main solvers for each type of optimization model:

- **LP** **CPLEX**
- **MIP, RMIP** **CPLEX**
- **NLP** **BARON, MINOS, CONOPT, ANTIGONE**
- **MINLP** **BARON, DICOPT, ANTIGONE**
Structure of a GAMS Model

SETS
Declaration
Assignment of members

PARAMETERS, TABLES, SCALARS (DATA)
Declaration
Assignment of values

VARIABLES
Declaration
Assignment of type
Assignment of bounds and/or initial values (optional)

EQUATIONS
Declaration
Definition

MODEL and SOLVE statements

OPTION for output file and solver

DISPLAY statements (optional)
Types of variables:

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Allowed Range of Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREE</td>
<td>$-\infty$ to $+\infty$</td>
</tr>
<tr>
<td>POSITIVE</td>
<td>0 to $+\infty$</td>
</tr>
<tr>
<td>NEGATIVE</td>
<td>$-\infty$ to 0</td>
</tr>
<tr>
<td>BINARY</td>
<td>0 or 1</td>
</tr>
<tr>
<td>INTEGER</td>
<td>0, 1, ..., 100</td>
</tr>
</tbody>
</table>

The default type is **FREE**

Remark: The variable being optimized must be **FREE** and not indexed
The keyword **EQUATIONS** is for **listing the names** (which are random) of the constraints and objective function.

The equations are **defined** by: equation_name..

**Syntax for the equality and inequality signs:**

- \( \text{=}E\text{=} \) for \( = \)
- \( \text{=}G\text{=} \) for \( \geq \)
- \( \text{=}L\text{=} \) for \( \leq \)

The basic arithmetic operators are:

- \( + \) addition
- \( - \) subtraction
- \( * \) multiplication
- \( / \) division
- \( ** \) exponent

**Example:** equation_name.. variable_1 + variable_2 =E= 1;

*The semi-colon ; is needed at the end of each equation.*
Bounds and initial values can be provided by adding a suffix to the variables.

**Syntax for specifying bounds and initial values:**

(variable name).LO lower bound (e.g., x.LO)
(variable name).UP upper bound (e.g., x.UP)
(variable name).FX fixed value
(variable name).L level value, meaning actual value (initial or final)
(variable name).M dual prices, Lagrange or Kuhn-Tucker multipliers

No need to specify lower bounds of zero for variables defined as **POSITIVE VARIABLE**.

In general, it is not a requirement to specify initial values for the variables.

However, for **nonlinear** models it is often advisable to provide an **initial guess** (e.g., X.L=4;).
**Keywords**

**MODEL**

The keyword **MODEL** is used to name the optimization model and to specify which equations are included in it.

\[
\text{MODEL (model name) /ALL/;}
\]

**OPTION**

The keyword **OPTION** is used to suppress output for debugging the compilation of the equations, and to set options for solvers (max CPU s).

\[
\begin{align*}
\text{OPTION} & \quad \text{(for debugging)} \\
\text{MODEL (model name) /eq1,eq2/;} & \quad \text{(set options for solvers)}
\end{align*}
\]

**SOLVE**

The keyword **SOLVE** calls the optimization solver. The syntax is as follows:

\[
\text{SOLVE (model name) USING (solver type) MINIMIZING (objective variable);}
\]

**DISPLAY**

The keyword **DISPLAY** shows the values of the requested symbols.

\[
\begin{align*}
\text{DISPLAY (variable name).suffix;} & \quad \text{and} \quad \text{DISPLAY (parameter or set name);} \\
\end{align*}
\]
GAMS
Example Problem 1
Transportation problem
### A Transportation Problem

**Objective**: minimize transportation cost

**Subject to**: demand satisfaction & supply constraints

<table>
<thead>
<tr>
<th>PLANTS</th>
<th>MARKETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Diego</td>
<td>New York</td>
</tr>
<tr>
<td>Seattle</td>
<td>Topeka</td>
</tr>
<tr>
<td>Seattle</td>
<td>Chicago</td>
</tr>
</tbody>
</table>

Transportation costs → $90/case/kMile

<table>
<thead>
<tr>
<th>Plant</th>
<th>Market 1</th>
<th>Market 2</th>
<th>Market 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Diego</td>
<td>325</td>
<td>275</td>
<td>300</td>
</tr>
<tr>
<td>Seattle</td>
<td>275</td>
<td>300</td>
<td>600</td>
</tr>
</tbody>
</table>

Transportation costs → $90/case/kMile
Indices (or sets):

- \( i \) : plants (San Diego, Seattle)
- \( j \) : markets (New York, Topeka, Chicago)

Given Data (or parameters):

- \( a_i \) : supply of commodity of plant \( i \) (in cases)
- \( b_j \) : demand for commodity at market \( j \) (cases)
- \( c_{ij} \) : transportation unit cost between plant \( i \) and market \( j \) ($/case)
A Transportation Problem

<table>
<thead>
<tr>
<th>Plants</th>
<th>Markets</th>
<th></th>
<th></th>
<th></th>
<th>supply</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New York</td>
<td>Chicago</td>
<td>Topeka</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seattle</td>
<td>2.5</td>
<td>1.7</td>
<td>1.8</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td>San Diego</td>
<td>2.5</td>
<td>1.8</td>
<td>1.4</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>demand</td>
<td>325</td>
<td>300</td>
<td>275</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Transportation costs → $90/case/kMile.

\[ c_{ij} = \frac{90 \cdot d_{ij}}{1000} \]
A Transportation Problem

Decision Variables:
\[ x_{ij} \]: amount to transfer from plant \( i \) to market \( j \). \( x_{ij} \geq 0, \quad \forall i, j \)

Objective Function:
Minimize \[ \sum_{i} \sum_{j} c_{ij} x_{ij} \text{ (}$K\text{)} \]

Constraints:
Supply limit in plant \( i \):
\[ \sum_{j} x_{ij} \leq a_{i}, \quad \forall i \]

Satisfy demand at market \( j \):
\[ \sum_{i} x_{ij} \leq b_{j}, \quad \forall j \]
Sets
  i  plants / seattle, san-diego /
  j  markets / new-york, chicago, topeka /

Parameters
  a(i)  capacity of plant i in cases / seattle 350, san-diego 600 /
  b(j)  demand at market j in cases / new-york 325
        chicago 300
        topeka 275 /

Table
  d(i,j)  distance in thousands of miles
        new-york               chicago      topeka
  seattle         2.5            1.7          1.8
  san-diego       2.5            1.8          1.4 /

Scalar
  f  freight in dollars per case per thousand miles /90/ ;

Parameter
  c(i,j)  transport cost in thousands of dollars per case
          c(i,j) = f*d(i,j)/1000;

Variables
  x(i,j)  shipment quantities in cases
  z  total transportation costs in thousands of dollars ;

Positive Variable
  x ;  z.lo = 1;

Equations
  cost  define objective function
  supply(i)  observe supply limit at plant i
  demand(j)  satisfy demand at market j

  cost .. z  =e=  sum(i,j, c(i,j)*x(i,j)) ;

  supply(i) .. sum(j, x(i,j))  =l=  a(i) ;

  demand(j) .. sum(i, x(i,j))  =g=  b(j) ;

Model
  transport /all/ ;

Solve
  transport Using LP Minimizing z;
Display
  x.L, x.M;
General notes

- Declare sets and variables first! You cannot refer to something that has not been defined!
- Terminate statements with semi-colons (;)
- GAMS compiler is not case-sensitive
- Lines starting with * are comment lines
- Names must start with a letter
- Descriptive text must:
  - fit on one line, and be no more than 80 characters long
  - not start with GAMS’ reserved words or contain the symbols =, ;/
Sets

- **Sets in GAMS ≡ indices in algebraic models, e.g.,**
  ```gams
  Sets
  i   canning plants / seattle, san-diego /,
  j   markets       / new-york, chicago, topeka / ;
  ```

- **Multi-words not allowed:** NEW-YORK not NEW YORK

- Can also write as:
  ```gams
  Set
  i   canning plants / seattle, san-diego /,
  j   markets       / new-york, chicago, topeka / ;
  ```

- **Use of the asterisk in set assignment:**
  ```gams
  Set
  T   time periods / 1991*2000 /,
  M   machines     / MACH1*MACH24 /;
  ```

  This corresponds to $T = \{1991, 1992, \ldots, 2000\}$ and $M = \{MACH1, MACH2, \ldots, MACH24\}$

- **ALIAS (I, IP); defines the SET (index) IP identical to the SET I**
Input Data

- **Entry by lists:**
  
  Parameters $a(i)$ capacity of plant $i$ in cases  
  / seattle 350  
  san-diego 600 /
  
  $b(j)$ demand at market $j$ in cases  
  / new-york 325  
  chicago 300  
  topeka 275 /

- **Entry by Tables:**
  
  Table $d(i,j)$ distance in thousands of miles  
  
  new-york | chicago | topeka  
  seattle 2.5 | 1.7 | 1.8  
  san-diego 2.5 | 1.8 | 1.4

  Alternatively: $d(\text{"seattle","new-york")}=2.5$; etc.

- **Entry by Direct Assignment:**
  
  Parameter $c(i,j)$ transport cost in thousands of dollars per case  
  $c(i,j)=f*d(i,j)/1000$;

- **Zero is default value for all un-assigned parameters and scalars**
Equations

- Note: equations must be declared and then defined

- Use of SUM and PROD operators:

  \[
  \text{SUM}(j, x(j)) \quad \longleftrightarrow \quad \sum_j x_j \\
  \text{PROD}(j, x(j)) \quad \longleftrightarrow \quad \prod_j x_j \\
  \text{SUM}((i,j), x(i,j)) \text{ or } \text{SUM}(i, \text{SUM}(j, x(i,j))) \quad \longleftrightarrow \quad \sum_i \sum_j x_{ij}
  \]

  Also, they can be used in direct assignment of PARAMETERS and SCALARS, e.g.,

  \[
  \text{SCALAR} \quad \text{TOTSUPPLY} \quad \text{total supply over all plants;}
  \]

  \[
  \text{TOTSUPPLY} = \text{SUM}(i, b(i));
  \]
The Dollar Operator ($)  

Provides a concise exception-handling capability

- **Example 1:** If \( y \geq 1.5 \), then \( x = 2 \), else \( x = 1 \)
  
  ```
  SCALAR X,Y;
  Y = 2; X = 1;
  X = 2\$(Y GE 1.5);
  ```

- **Example 2:** If \( y \leq 1.5 \), then \( x = 2 \), else \( x = 1 \)
  
  ```
  SCALAR X,Y;
  Y = 2; X = 1;
  X\$(Y LE 1.5) = 2;
  ```

- **Example 3:** If \( x_i \neq 0 \), then \( \rho_i = 1/x_i \), else \( \rho_i = 0 \)
  
  ```
  rho(i) = (1/x(i))\$(x(i) NE 0);
  ```

- **Example 4 (Equations):** \( j = \{1,2,3,4\} \).
  \[
  \sum_{j=2}^{4} x_j = 1
  \]
  
  ```
  EQ1.. SUM(j$ (j>1),x(j)) =E= 1;
  ```

- **Example 5 (Equations):** If \( x_i \neq 0 \), then \( z_i = y_i - 3 \)
  
  ```
  EQ2$(x(i) NE 0).. z(i) =E= y(i) - 3;
  ```
GAMS Compilation

- Open **GAMS IDE**
- Create a **GAMS project file** *(project name)*.gpr
- Create a **GAMS input file** *(file name)*.gms and save it in the same directory with the project file
- Run GAMS by pressing **F9** or the **run button** in the GAMS IDE
- After the **compilation** of the .gms file, an output file is created *(file name)*.lst
The main elements of the GAMS output are:

- Echo Print
- Error Messages
- Equation Listings
- Model Statistics
- Status Reports
- Solution Reports
The input statement `DISPLAY x.L, x.M;`

--- 65 VARIABLE x.L shipment quantities in cases

<table>
<thead>
<tr>
<th>City</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>new-york</td>
<td></td>
</tr>
<tr>
<td>chicago</td>
<td></td>
</tr>
<tr>
<td>topeka</td>
<td></td>
</tr>
<tr>
<td>seattle</td>
<td>300.000</td>
</tr>
<tr>
<td>san-diego</td>
<td>325.000</td>
</tr>
</tbody>
</table>

--- 65 VARIABLE x.M shipment quantities in cases

<table>
<thead>
<tr>
<th>City</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>new-york</td>
<td></td>
</tr>
<tr>
<td>chicago</td>
<td></td>
</tr>
<tr>
<td>topeka</td>
<td></td>
</tr>
<tr>
<td>seattle</td>
<td>EPS 0.036</td>
</tr>
<tr>
<td>san-diego</td>
<td>0.009</td>
</tr>
</tbody>
</table>
Final Remarks

Some other useful operators are the following:

- Dollar operator (\$): it can be used to restrict the elements of a set; roughly speaking, it has the function of command IF.

- **LOOP** keyword which can be used to perform repetitive calculations for parameters or sets.

- **FOR** can be used for multiple **SOLVE** statements in an iterative algorithm.

- There are also other usual programming commands; e.g., **IF** and **WHILE**.

GAMS is available for several operating systems (Windows, Linux, Mac, Solaris).

GAMS is free, but the demo version could solve problems of limited size.

Solvers are NOT free (license is needed for every solver).

Other Algebraic Modeling Languages: **AMPL**, **AIMMS**, etc.
A maximum-profit production problem

A furniture company wants to maximize its profits from the manufacture of four different types of desks.

Each different desk type requires different number of man-hours in each of the company sections, carpentry and finishing.

The profit per unit of every different type of desk sold is different and given.

The capacity of man-hours for carpentry and finishing is also given.

Formulate the optimization problem, and then introduce it to GAMS and solve it.
Problem Formulation

Sets:  
- $i$: desk type (d1, d2, d3, d4)
- $j$: sector (carpentry, finishing)

Data:  
- $\text{caplim}(j)$: capacity limit of sector $j$
- $\text{profit}(i)$: profit from selling desk type $i$
- $\text{labor}(j,i)$: labor requirements (man-hours)

Decision Variable:  
- $x_i$: number of desks $i$ produced

Objective Function:  
- $\sum_i x_i profit_i$

Constraint:  
- $\sum_i \text{labor}_{j,i} x_i \leq \text{caplim}_j \forall j$
Sets  
desk / d1, d2, d3, d4 /
    shop / carpentry, finishing /

Table labor(shop,desk)  labor requirements (man-hours)

        d1   d2   d3   d4

carpentry  4    9    7   10
finishing   1    1    3   40

Parameters

caplim(shop)  capacity (man hours) / carpentry = 6000, finishing = 4000 /
price(desk)   per unit sold ($) / d1 = 12, d2 = 20, d3 = 15, d4 = 40 /

Variables

    mix(desk)  mix of desks produced (number of desks)
profit      total profit  ( $ )

Positive Variable mix

Equations

    cap(shop)  capacity constraint (man-hours)
ap         accounting: total profit ( $ );

    cap(shop)..  sum (desk, labor(shop,desk) * mix(desk)) =l= caplim(shop);

    ap..  profit  =e= sum(desk, price(desk) * mix(desk));

Model pmp product mix problem / all /; Solve pmp maximizing profit using lp;
• GAMS User Guide
• Expanded GAMS Guide (McCarl)
• Example from Model Library

![GAMS Model Library](image)

Example 1

A Transportation Problem (TRNSPORT, SEQ=1)

This problem finds a least cost shipping schedule that meets requirements at markets and supplies at factories.
Consider this problem

A “State-Task Network”

React

C

Add

Blend 1

Blend 2

P1

P2

320 t

3h

400t

80%

20%

125 t

75%

25%

1h

600t

1450t

How long to make these quantities?

“A State-Task Network”

“A “state”

“task”

A

B

“state”

“task”

3h

400t

80%

20%

125 t

75%

25%

1h

600t

1450t

How long to make these quantities?
Answer: 26 hours

Batches of P2

Batches of P1
Will a bioprocess design work?

Interconnected process steps

Complex purification

Many equipment items
Yes, but sensitive to chromatography operation
Can this site make enough of these new pharmaceuticals?

Yes, but these resources are very busy!
Conclusions

- We are working together with developing country manufactures to improve their vaccine production processes;

- We are designing and implementing new vaccine production technologies;

- We are doing multi-level modelling using the following tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).