Obtaining Simultaneous Equation Models through a unified shared-memory scheme of metaheuristics

Francisco Almeida
Departamento de Estadística, Investigación Operativa y Computación, Universidad de La Laguna

Domingo Giménez
Departamento de Informática y Sistemas, Universidad de Murcia

Jose Juan López Espín
Centro de Investigación Operativa, Universidad Miguel Hernández
Contents

1 Motivation
2 Obtaining SEM
3 Parametrized metaheuristics
4 Unified shared-memory metaheuristics
5 Experiments
6 Conclusions
Simultaneous Equation Models

- Simultaneous Equation Models (SEM) have been used in econometrics for years (Keynes model). They are used in medicine, network simulation, study of sociological behavior, etc.
- Traditionally, SEM have been developed by people with a wealth of experience in the particular problem represented by the model.
- Our objective is to develop an algorithm which, given a set of values of the variables, finds a satisfactory SEM.
- The space of the possible solutions is very large and exhaustive search methods are not suitable here.
- Our work is on the application of metaheuristics.
Simultaneous Equation Models (SEM) have been used in econometrics for years (Keynes model). They are used in medicine, network simulation, study of sociological behavior, etc.

Traditionally, SEM have been developed by people with a wealth of experience in the particular problem represented by the model.

Our objective is to develop an algorithm which, given a set of values of the variables, finds a satisfactory SEM.

The space of the possible solutions is very large and exhaustive search methods are not suitable here.

Our work is on the application of metaheuristics.
Simultaneous Equation Models

- Simultaneous Equation Models (SEM) have been used in econometrics for years (Keynes model). They are used in medicine, network simulation, study of sociological behavior, etc.

- Traditionally, SEM have been developed by people with a wealth of experience in the particular problem represented by the model.

- Our objective is to develop an algorithm which, given a set of values of the variables, finds a satisfactory SEM.

- The space of the possible solutions is very large and exhaustive search methods are not suitable here.

- Our work is on the application of metaheuristics.
Simultaneous Equation Models

- Simultaneous Equation Models (SEM) have been used in econometrics for years (Keynes model). They are used in medicine, network simulation, study of sociological behavior, etc.
- Traditionally, SEM have been developed by people with a wealth of experience in the particular problem represented by the model.
- Our objective is to develop an algorithm which, given a set of values of the variables, finds a satisfactory SEM.
- The space of the possible solutions is very large and exhaustive search methods are not suitable here.
- Our work is on the application of metaheuristics.
Simultaneous Equation Models

- Simultaneous Equation Models (SEM) have been used in econometrics for years (Keynes model). They are used in medicine, network simulation, study of sociological behavior, etc.
- Traditionally, SEM have been developed by people with a wealth of experience in the particular problem represented by the model.
- Our objective is to develop an algorithm which, given a set of values of the variables, finds a satisfactory SEM.
- The space of the possible solutions is very large and exhaustive search methods are not suitable here.
- Our work is on the application of metaheuristics.
Parametrized metaheuristics

- To tune a metaheuristic to a problem, experiments with several parameters (intra-metaheuristic parameters) and functions
- To obtain a good metaheuristic for a problem, experiments with several metaheuristics

We propose the use of unified parametrized schemes for metaheuristics: different values of inter-metaheuristic parameters would provide different metaheuristics or hybridation/combination of metaheuristics
Parametrized metaheuristics

- To tune a metaheuristic to a problem, experiments with several parameters (intra-metaheuristic parameters) and functions
- To obtain a good metaheuristic for a problem, experiments with several metaheuristics

We propose the use of unified parametrized schemes for metaheuristics: different values of inter-metaheuristic parameters would provide different metaheuristics or hybridation/combination of metaheuristics
Parametrized metaheuristics

- To tune a metaheuristic to a problem, experiments with several parameters (intra-metaheuristic parameters) and functions
- To obtain a good metaheuristic for a problem, experiments with several metaheuristics

⇒

We propose the use of **unified parametrized schemes for metaheuristics**: different values of inter-metaheuristic parameters would provide different metaheuristics or hybridation/combination of metaheuristics
Parallel-parametrized metaheuristics

- To select a satisfactory metaheuristic and to tune it to the problem requires a lot of experiments.
- When applying metaheuristics to obtain satisfactory SEM a large number of systems are solved.

We propose the use of unified parallel-parametrized schemes for metaheuristics: the different metaheuristics obtained from the parametrized scheme are parallelized together, with parallel parameters for optimization of the execution time.
Parallel-parametrized metaheuristics

- To select a satisfactory metaheuristic and to tune it to the problem requires a lot of experiments.
- When applying metaheuristics to obtain satisfactory SEM a large number of systems are solved.

We propose the use of **unified parallel-parametrized schemes for metaheuristics**: the different metaheuristics obtained from the parametrized scheme are parallelized together, with parallel parameters for optimization of the execution time.
Parallel-parametrized metaheuristics

- To select a satisfactory metaheuristic and to tune it to the problem requires a lot of experiments.
- When applying metaheuristics to obtain satisfactory SEM a large number of systems are solved.

We propose the use of **unified parallel-parametrized schemes for metaheuristics**: the different metaheuristics obtained from the parametrized scheme are parallelized together, with parallel parameters for optimization of the execution time.
SEM: a system with $N$ equations, $N$ endogenous variables, $K$ exogenous variables and sample size $d$ is:

\begin{align*}
y_1 &= \gamma_{1,1}x_1 + \cdots + \gamma_{1,K}x_K + \beta_{1,2}y_2 + \beta_{1,3}y_3 + \cdots + \beta_{1,N}y_N + u_1 \\
y_2 &= \gamma_{2,1}x_1 + \cdots + \gamma_{2,K}x_K + \beta_{2,1}y_1 + \beta_{2,3}y_3 + \cdots + \beta_{2,N}y_N + u_2 \\
&\quad \vdots \\
y_N &= \gamma_{N,1}x_1 + \cdots + \gamma_{N,K}x_K + \beta_{N,1}y_1 + \cdots + \beta_{N,N-1}y_{N-1} + u_N
\end{align*}

The problem: given values of $x_i$ and $y_i$ (vectors of dimension $d$, obtained by experimentation or survey), obtain the system ($\beta_{i,j}$ and $\gamma_{i,j}$ non equal to zero) which best represents the variables’ dependencies, according to some criterion.
Methods to solve SEM:

- Maximum Likelihood
- Indirect Least Square
- **Two-Steps Least Square (2SLS)**
- Three-Steps Least Square

We use 2SLS:

- Lower computational cost \(O(NK^2d)\)
- Can be applied in more cases

... but the conclusions about the application of metaheuristics do not depend on the method used.
Methods to solve SEM:

- Maximum Likelihood
- Indirect Least Square
- **Two-Steps Least Square (2SLS)**
- Three-Steps Least Square

We use 2SLS:

- Lower computational cost \( O(NK^2d) \)
- Can be applied in more cases

... but the conclusions about the application of metaheuristics do not depend on the method used.
Methods to solve SEM:
- Maximum Likelihood
- Indirect Least Square
- **Two-Steps Least Square (2SLS)**
- Three-Steps Least Square

We use 2SLS:
- Lower computational cost ($O(NK^2d)$)
- Can be applied in more cases

... but the conclusions about the application of metaheuristics do not depend on the method used.
For each variable it should be decided if it is included (1) or not (0) in the system: number of possible models $2^{N(N+K)}$.

Different criteria can be used to determine the goodness of the model. We use the Akaike Information Criterion:

$$d \cdot \ln |\hat{\Sigma}_e| + 2 \sum_{i=1}^{N}(n_i + k_i - 1) + N^2 + N$$

where $|\hat{\Sigma}_e|$ is the determinant of the error covariance matrix, and $e_i$ represents the difference between $y_i$ and its estimation.

... but the conclusions about the application of metaheuristics do not depend on the criterion used.
For each variable it should be decided if it is included (1) or not (0) in the system: number of possible models $2^{N(N+K)}$.

Different criteria can be used to determine the goodness of the model. We use the Akaike Information Criterion:

$$d \cdot \ln|\hat{\Sigma}_e| + 2 \sum_{i=1}^{N} (n_i + k_i - 1) + N^2 + N$$

where $|\hat{\Sigma}_e|$ is the determinant of the error covariance matrix, and $e_i$ represents the difference between $y_i$ and its estimation.

... but the conclusions about the application of metaheuristics do not depend on the criterion used.
For each variable it should be decided if it is included (1) or not (0) in the system: number of possible models $2^{N(N+K)}$

Different criteria can be used to determine the goodness of the model. We use the **Akaike Information Criterion**:

$$d \cdot \ln |\hat{\Sigma}_e| + 2 \sum_{i=1}^{N} (n_i + k_i - 1) + N^2 + N$$

where $|\hat{\Sigma}_e|$ is the determinant of the error covariance matrix, and $e_i$ represents the difference between $y_i$ and its estimation.

... but the conclusions about the application of metaheuristics do not depend on the criterion used.
Objectives

- To obtain a tool to efficiently apply and tune metaheuristics for the SEM problem
  Done: applied to model the preeclampsia, economic data...
- So, we develop unified parametrized schemes of metaheuristics
  Done: GRASP, Genetic Algorithms, Scatter Search and combinations/hybridations
- and unified parallel-parametrized schemes
  Done: in OpenMP for shared memory, with parametrized parallel functions
- Future:
  - More metaheuristics
  - Parallel schemes for other systems: message-passing and GPU
  - Hyperheuristics: autonomous selection of adequate values for the metaheuristic parameters
  - Auto-optimization: autonomous selection of adequate values for the parallelism parameters
Objectives

- To obtain a tool to efficiently apply and tune metaheuristics for the SEM problem
  Done: applied to model the preeclampsia, economic data...

- So, we develop unified parametrized schemes of metaheuristics
  Done: GRASP, Genetic Algorithms, Scatter Search and combinations/hybridations

- and unified parallel-parametrized schemes
  Done: in OpenMP for shared memory, with parametrized parallel functions

- Future:
  - More metaheuristics
  - Parallel schemes for other systems: message-passing and GPU
  - Hyperheuristics: autonomous selection of adequate values for the metaheuristic parameters
  - Auto-optimization: autonomous selection of adequate values for the parallelism parameters
Objectives

- To obtain a tool to efficiently apply and tune metaheuristics for the SEM problem
  Done: applied to model the preeclampsia, economic data...

- So, we develop unified parametrized schemes of metaheuristics
  Done: GRASP, Genetic Algorithms, Scatter Search and combinations/hybridations

- and unified parallel-parametrized schemes
  Done: in OpenMP for shared memory, with parametrized parallel functions

- Future:
  - More metaheuristics
  - Parallel schemes for other systems: message-passing and GPU
  - Hyperheuristics: autonomous selection of adequate values for the metaheuristic parameters
  - Auto-optimization: autonomous selection of adequate values for the parallelism parameters
Objectives

- To obtain a tool to efficiently apply and tune metaheuristics for the SEM problem
  Done: applied to model the preeclampsia, economic data...
- So, we develop unified parametrized schemes of metaheuristics
  Done: GRASP, Genetic Algorithms, Scatter Search and combinations/hybridations
- and unified parallel-parametrized schemes
  Done: in OpenMP for shared memory, with parametrized parallel functions
- Future:
  - More metaheuristics
  - Parallel schemes for other systems: message-passing and GPU
  - Hyperheuristics: autonomous selection of adequate values for the metaheuristic parameters
  - Auto-optimization: autonomous selection of adequate values for the parallelism parameters
Unified metaheuristic scheme

Initialize($S$)

while (not EndCondition($S$))

$SS = \text{Select}(S)$
$SS1 = \text{Combine}(SS)$
$SS2 = \text{Improve}(SS1)$
$S = \text{Include}(SS2)$

Facilitates working with different metaheuristics by reusing functions
The use of inter-metaheuristic parameters facilitates work with different metaheuristics/hybridation/combination by selecting different values of the parameters in the functions.
Metaheuristics

- Pure metaheuristics: GRASP, Genetic algorithms (GA), Scatter search (SS)
- Combinations: GRASP+GA, GRASP+SS, GA+SS, GRASP+GA+SS
Initialize

- Randomly generate valid elements. The number of elements ($INEIni$) varies to tune the metaheuristic to the problem (intra parameter), but different values can correspond to different metaheuristics (inter-metaheuristic parameter).

- Generated elements can be improved with local search, greedy..., with a percentage of elements to improve ($PEIIni$) and an intensification in the improvement ($IIEIni$).

- A number of elements ($NERIni$) is selected to form the reference set.

$$ParamIni = (INEIni, PEIIni, IIEIni, NERIni)$$
Initialize

- Randomly generate valid elements. The number of elements (%INEIni%) varies to tune the metaheuristic to the problem (intra parameter), but different values can correspond to different metaheuristics (inter-metaheuristic parameter).

- Generated elements can be improved with local search, greedy..., with a percentage of elements to improve (%PEIni%) and an intensification in the improvement (%IIIni%).

- A number of elements (%NERIni%) is selected to form the reference set.

$$ParamIni = (%INEIni, %PEIni, %IIIni, %NERIni)$$
Initialize

- Randomly generate valid elements. The number of elements ($INEin$) varies to tune the metaheuristic to the problem (intra parameter), but different values can correspond to different metaheuristics (inter-metaheuristic parameter).

- Generated elements can be improved with local search, greedy..., with a percentage of elements to improve ($PEIlIni$) and an intensification in the improvement ($IIEin$).

- A number of elements ($NERIni$) is selected to form the reference set.

\[ ParamIni = (INEin, PEIlIni, IIEIni, NERIni) \]
In the context of obtaining SEM, a key aspect involves the selection of parameters. The process typically involves:

- Selecting the best and worst elements according to some function (fitness function, scatter function...) are selected.
- The number of best elements is $NBESel$, and the number of worst $NWESel$, and normally $NBESel + NWESel = NERIni$.

The parameters selected are:

$$ParamSel = (NBESel, NWESel)$$
Combine

- A certain number of combinations between the best elements ($NBBCom$),
  between the best and the worst ($NBWCom$),
  and between the worst ($NWWCom$).

$$ParamCom = (NBBCom, NBWCom, NWWCom)$$
A percentage of elements (\(PEllmp\)) are improved by local search..., with a certain intensification (\(IIEImp\)).

A percentage of elements which are “distant” (\(PEDImp\)) from the reference set are generated, and an improvement is applied to these elements with a certain intensification (\(IIDImp\)).

\[
\text{ParamImp} = (PEllmp, IIEImp, PEDImp, IIDImp)
\]
A percentage of elements ($PEImp$) are improved by local search..., with a certain intensification ($IIEImp$).

A percentage of elements which are “distant” ($PEImp$) from the reference set are generated, and an improvement is applied to these elements with a certain intensification ($IIDImp$).

$$ParamImp = (PEImp, IIEImp, PEDImp, IIDImp)$$
Include and EndCondition

- The best $NBEInc$ elements are included in the reference set, and the others ($NERIni - NBEInc$) are the most “distant” from the best ones according to some distance function

$$ParamInc = (NBEInc)$$

- The method converges when a maximum number of iterations ($MNIEnd$) or a maximum number of iterations without improving the best solution ($NIREnd$) is performed

$$ParamEnd = (MNIEnd, NIREnd)$$

(they could be considered inter-metaheuristic parameters)
The best $NBE_{Inc}$ elements are included in the reference set, and the others ($NER_{Ini} - NBE_{Inc}$) are the most “distant” from the best ones according to some distance function

$$Param_{Inc} = (NBE_{Inc})$$

The method converges when a maximum number of iterations ($MNI_{End}$) or a maximum number of iterations without improving the best solution ($NIRE_{End}$) is performed

$$Param_{End} = (MNI_{End}, NIRE_{End})$$

(they could be considered inter-metaheuristic parameters)
Parallel-parametrized scheme

Initialize($S, \text{ParamIni}, \text{ThreadsIni}$)  
\textbf{while} (not EndCondition($S, \text{ParamEnd}, \text{ThreadsEnd}$))  
  \begin{align*}  
  SS &= \text{Select}(S, \text{ParamSel}, \text{ThreadsSel}) \\  SS1 &= \text{Combine}(SS, \text{ParamCom}, \text{ThreadsCom}) \\  SS2 &= \text{Improve}(SS1, \text{ParamImp}, \text{ThreadsImp}) \\  S &= \text{Include}(SS2, \text{ParamInc}, \text{ThreadsInc})  
\end{align*}

Independent parallelization of the functions, with \textit{parallelism parameters} (number of threads) for each function. The optimum value of the \textit{parallelism parameters} depends on the values of the \textit{metaheuristic parameters} (the metaheuristic or combination of metaheuristics)
Identify functions with the same parallel scheme:

Loop parallelism

```c
omp_set_num_threads(one-loop-threads)
#pragma omp parallel for
loop in elements
treat element
```

i.e.: Initialize, Combine...
Nested parallelism

```c
omp_set_num_threads(first-level-threads)
#pragma omp parallel for
loop in elements
treat-element-second-level(first-level-threads)

treat-element-second-level(first-level-threads):
  omp_set_num_threads(second-level-threads(first-level-threads))
#pragma omp parallel for
loop in elements
treat element
does, i.e.: Initialize, Improve...
```

Allows fine and coarse grained parallelism by changing the number of threads in each level
Initialize($S, INEIni, PEIni, IIEIni, NERIni,$
\quad ntIni, ntIni1, ntIni2)$

\textbf{while} (not EndCondition($S, NMIEnd, NIREnd$))

$SS = \text{Select}(S, NBESel, NWESel)$

$SS1 = \text{Combine}(SS, NBBCom, NBWCom, NWWCom,$
\quad ntCom, ntEva)$

$SS2 = \text{Improve}(SS1, PEImp, IIEImp, PEDImp, IIDImp,$
\quad ntImp1, ntImp2, ntMut1, ntMut2)$

$S = \text{Include}(SS2, NBEincc,$
\quad ntInc)$
Computational systems

- **Nodes of Rosebud (cluster of the Polytechnic University of Valencia):**
  - Rosebud4: Intel Core 2 Quad.
  - Rosebud8: Fujitsu Primergy RXI600 with 4 Dual-Core processors Intel Itanium2.

- **Ben–Arabí (Supercomputing Center of Murcia):**
  - Ben: HP Integrity Superdome SX2000 with 128 cores of the processor Intel Itanium-2 dual-core Montvale.
  - Arabí: cluster of 102 nodes, each one with 8 cores of the processor Intel Xeon Quad-Core L5450 (one node used).
## Parameters

Satisfactory values for the “pure” metaheuristics obtained by experimentation

<table>
<thead>
<tr>
<th></th>
<th>GRASP</th>
<th>GA</th>
<th>SS</th>
<th>GRASP+GA</th>
<th>GRASP+SS</th>
<th>GA+SS</th>
<th>GRASP+GA+SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INEIni</td>
<td>200</td>
<td>500</td>
<td>100</td>
<td>200</td>
<td>200</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>NERIni</td>
<td>-</td>
<td>500</td>
<td>20</td>
<td>200</td>
<td>20</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>PEIIni</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IIEIni</td>
<td>10</td>
<td>-</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>NBESel</td>
<td>-</td>
<td>500</td>
<td>10</td>
<td>200</td>
<td>10</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>NWESel</td>
<td>-</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>NBBCom</td>
<td>-</td>
<td>250</td>
<td>90</td>
<td>100</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>NBWCom</td>
<td>-</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>NWWCom</td>
<td>-</td>
<td>-</td>
<td>90</td>
<td>-</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>PEIImp</td>
<td>-</td>
<td>0</td>
<td>100</td>
<td>-</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IIEImp</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>PEDImp</td>
<td>-</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>IIDImp</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>NBEInc</td>
<td>-</td>
<td>500</td>
<td>10</td>
<td>200</td>
<td>10</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>
The results previously obtained with Genetic algorithms are improved

Value of AIC in successive iterations, $N = K = 20$

$d = 100,$  \hspace{1cm}  $d = 200$
Statistical study of the influence of the parameters can be carried out.

700 combined metaheuristics are generated, 100 in the environment of each basic metaheuristic in the space of metaheuristics.
Motivation

Obtaining SEM

Parametrized metaheuristics

Unified shared-memory metaheuristics

Experiments

Conclusions

Speed-up obtained for different metaheuristics in Rosebud.

\[ N = K = 20, \ d = 100 \]
Use of parallelism parameters

speed-up with the maximum number of cores, the optimum number of threads for the complete program, and different numbers of threads in the different functions and levels

\[ N = K = 20, \quad d = 100 \]
The use of parameterized metaheuristics allows us:

- To experiment with different metaheuristics and combinations (inter-parameters) to obtain a satisfactory one for a particular problem
- To experiment with different parameters (intra-parameters) to tune the metaheuristic to the problem
- To develop unified parallel schemes, which can be optimized by selecting the parallel parameters (number of threads in the different functions) for the particular metaheuristic and problem

The methodology is being applied to other problems: tasks-to-processors assignment, p-hub, signal filters design, electricity consumption in water wells
Future research

- Inclusion of more “pure” metaheuristics
- Design of hyperheuristics to automatically select the values of the inter-parameter for a particular problem
- Inclusion of auto-optimization in the parallel scheme, with some engine to autonomously select the number of threads
- Develop unified parallel schemes for other computational systems (message-passing, hybrid, GPU...)