Decomposition and High Throughput Solution of EP’s

Michael C. Ferris
University of Wisconsin
Tom Rutherford, Ann Arbor
So what, why bother, who cares?

- Motivate why we use grid computation
- Show how we implement it
  - GAMS
  - Condor
- Explain the benefits
- What are the limiting factors?
Walras meets Wardrop

What is the effect on housing prices of increasing capacity on the red arcs?
\[ T_{ik} \leq c_{ij}(F_{ij}) + T_{jk} \perp x_{ij}^k \geq 0 \]

\[ F_{ij} = \sum_k x_{ij}^k \]

\[ N_{jk} + \sum_i x_{ij}^k = \sum_i x_{ji}^k \]

\[ x_{ij}^k = \text{people travel (i,j) work at k} \]
\[ T_{ik} = \text{(shortest) time i to k} \]
\[ N_{ik} = \text{number live i work k} \]
\[ D_k(w_k) \leq N_{kk} + \sum_i x_{ik} \perp w_k \geq 0 \]

\[ V_{ik}(w_k, p_i, T_{ik}) \leq U \perp N_{ik} \geq 0 \]

\[ \bar{N} = \sum_{i,k} N_{ik} \]

\[ w_k = \text{labor cost at } k \]
\[ p_i = \text{housing price at } i \]
\[ N_{ik} = \text{number live i work k} \]
$T_{ik} \leq c_{ij}(F_{ij}) + T_{jk} \perp x_{ij}^k \geq 0$

$F_{ij} = \sum_k x_{ij}^k \perp F_{ij}$

- **Square**
  
- **Prices found**

- **Apply shocks**

- **Tolling**

- **Different housing types**

- **Different transportation types**

$D_k(w_k) \leq N_{kk} + \sum_i x_{ik}^k \perp w_k \geq 0$

$V_{ik}(w_k, p_i, T_{ik}) \leq U \perp N_{ik} \geq 0$

$\bar{N} = \sum_{i,k} N_{ik} \perp U$

$\sum_k N_{ik} H_{ik}(w_k, p_i) \leq \bar{H}_i \perp p_i \geq 0$
Stochastic Games

- Two players, infinite (discrete) time
- Prototype (Cournot) example on grid
  - (i,j) = players machines
- Investment and depreciation affects probabilities of changing state
- Variables are investment levels, quantities, prices
Sparsity pattern

<table>
<thead>
<tr>
<th>n</th>
<th>MCP size</th>
<th>LU-SOL</th>
<th>UMF PACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>3.2K</td>
<td>0.7</td>
<td>0.4</td>
</tr>
<tr>
<td>50</td>
<td>20K</td>
<td>8.5</td>
<td>2.8</td>
</tr>
<tr>
<td>100</td>
<td>80K</td>
<td>52.3</td>
<td>12.3</td>
</tr>
<tr>
<td>200</td>
<td>320K</td>
<td>479.0</td>
<td>56.4</td>
</tr>
</tbody>
</table>

Jacobian
Sparsity pattern

Jacobian

LU factors of Jacobian
A use for grid computation

- Enhance speed (or size) of computation model
  - Linear algebra
    - May not have LU exploitable structure
  - Decomposition approaches
    - Benders, Dantzig-Wolfe, Lagrangian
    - Jacobi, Gauss-Seidel
Trade/Policy Model (MCP)

- Split model (18,000 vars) via region
- Gauss-Seidel, Jacobi, Asynchronous
- 87 regional subprobs, 592 solves
Gauss-Seidel

loop(iter$(not done),
    loop(p,
        r(i) = yes$inp(i,p);
        x.fx(i)$(not r(i)) = x.l(i);
        solve trademod using mcp;
        x.lo(i) = 0; x.up(i) = xup(i) );
);
gemtap.solvelink = 3;
loop(iter$(not done),
    loop(p,
        r(i) = yes$inp(i,p);
x.fx(i)$(not r(i)) = x.l(i);
solve gemtap using mcp;
x.lo(i) = 0; x.up(i) = xup(i);
h(p) = gemtap.handle);

repeat; loop(p$h(p),
    if(handlestatus(h(p)),
        gemtap.handle = h(p); execute_loadhandle gemtap;
h(p) = 0; ););
until card(h) = 0;);
gemtap.solvelink = 3;
repeat; loop(p$(h(p)),
  if ( handlestatus(h(p)),
    gemtap.handle = h(p);
    execute_loadhandle gemtap;  h(p) = 0;
    if (sum(k, dev(k)) > tol,
      loop(k$(h(k) eq 0 and dev(k),
        gemtap.number = ord(k);
        r(i) = yes$inp(i,k);
        x.fx(i)$$(not r(i)) = x.l(i);
        solve gemtap using mcp;
        x.lo(i) = 0; x.up(i) = xup(i);
        h(k) = gemtap.handle;
      ));
    until (card(handle) = 0);
Condor as a Grid Computer

- Uses dedicated clusters and cycles from desktop workstations (> 1000 machines available for “ferris”)
- Heterogeneous machines, with or without shared file system
- Machines updated regularly
- Fault tolerance
  - Jobs submitted are eventually executed
- Available for download, configurable
Master-Worker Paradigm

- **Master**
  - Generates tasks
  - Responsible for synchronization
- **Worker**
  - Processes individual tasks
  - Reports back results
- **Simple! Dynamic! Fault tolerant!**
- **No worker inter-communication**
Worker or task

- Local copy of job needed
  - Zip file, job dir
  - Mimic environment
- Start flag
- End flag
- Trigger file
  - Updates
MW-GAMS

- Generate “gams task” worker
- Data common to all tasks is sent to workers only once
- (Try to) retain workers until the whole computation is complete—don’t release them after a single task
- Modeler just flips a switch!
  - Usecondor=mw,lnx,win,glow,sun
Trade/Policy Model (MCP)

• Split model (18,000 vars) via region

• Gauss-Seidel, Jacobi, Asynchronous

• 87 regional subprobs, 592 solves
Model knowledge decomposition

- Pink model - open economy (regions)
- Green model - (partial) spatial equilibrium (commodities)
- Links are imports and exports

Calibrate supply and demand functions to points, and communicate functional forms, not points
Deviations by iteration
GAMS/Grid

- Commercial modeling system – abundance of real life models to solve

- Any model types and solvers allowed
  - Scheduling problems
  - Radiotherapy treatment planning
  - World trade (economic) models
  - Sensitivity analysis
  - Cross validation, feature selection

- Little programming required
- Separation of model and solution maintained
Conclusions

• Grid systems available (e.g. Condor, IBM, SUN)
• Grid computing convenient via simple language extensions to modeling languages
• Can experiment with coarse grain parallel approaches for solving difficult problems
• Exploiting underlying structure and model knowledge key for “larger, faster” solution

• Easy, adaptive, improves, need expertise
Shortcomings/Future Work

• Iterative scheme updates small amount of model “data”
• As convergence occurs (prove it!) models become easy to solve (great start point)
• Model regeneration time is longer than solution time!