A parallel VNS scheme with ILP neighbourhoods. Application to a discrete Unit Commitment Problem

Nicolas DUPIN

University Bordeaux 1, EDF R&D-OSIRIS, PhD Thesis INRIA-EDF

META 2012, Port-El Kantaoui, october 2011, 29th
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Branch&Bound algorithm
Primal heuristics, using LP relaxation and branching with local search to provide integer solutions:

- **Feasibility Pump**: heuristic searching integer solution close to the LP solution. Useful to get first solutions.

- **RINS**: Fixing common integer values in the LP relaxation and the best solution known in a small ILP.

- **Diving heuristics**: Diving in the Branch&Bound tree, to find integer solutions.
Branch & Bound convergence for our problem

⇒ Efficient ILP heuristics on the first stages.
An ILP solver can be used to generate quickly good solutions.
For a time limit 15 min, it is more efficient to have fixed variables smartly.
Goal

Improve the primal bound convergence on our specific problem:

- Very good quality of the LP relaxation.
- Great efficiency of ILP heuristics.
- Is it possible to improve the convergence of these “blind” heuristics with some knowledge of the problem?
- Take advantage of IP heuristics and classical metaheuristics.

Variable Neighbourhood Search (VNS) algorithm adapted to that goal.
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Variable Neighbourhood Search (VNS) ideas

- Consider different types of neighbourhoods for a defined solution.
- Change systematically the neighbourhood within a local search.
- A local extrema of the VNS is a local extrema for all types of neighbourhoods.
- Better and fewer local extrema.
ILP neighbourhoods

Usually, VNS choose a solution in a given neighbourhoods. Our approach

- Fix a subset of the integer variables of the current solution.
- ILP Resolution limited in the Branch & Bound enumeration (number of nodes, time limit, stopping criteria with the gap lower - upper bound ...).
- Specialized parametrization of the ILP resolution. Depends on the expected size of the ILP post fixing.

⇒ The cost of the solution is given with the cost given by the ILP solver. The solution has at least the same cost than the original one (solution of this MIP).
Parallel VNS algorithm with ILP neighbourhoods

- Step 1: start with the current best solution.
- Step 2: Choose $n$ ILP neighbourhoods ($n$ number of threads).
- Step 3: Solve parallelly in a given time limit these ILP.
- Step 4: Take the best solution. If the stopping criteria is not met (number of iterations, time limit, number of iteration without improvement), go to step 1.

$\Rightarrow$ A steepest descent VNS algorithm.
Strategies for choosing ILP neighbourhoods

A lot of strategies for the choice of neighbourhoods:

- Deterministic sequences and/or random ILP neighbourhoods?
- Neighbourhoods with the same time limit parallely.
- On large ILP neighbourhood with a bigger time limit in parallel with a sequence of several smaller ILP neighbourhoods.
- Learning: stats to detect best types of neighbourhoods.
- small ILP neighbourhood till improvements, increase the size of neighbourhoods to get away from a local optimum . . .
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Useful functionalities

- **Aggregator**: detect and suppress the implied fixed variables or useless constraints, leading to a smaller MIP.
- **Warmstart**: ILP Resolution easier with an initial solution given to the solver.
How to parametrize an ILP solver?

- General mode mipemphasis to feasibility, to activate at most ILP heuristics.
- Node limit / time limit.
- Limit cutting plane passes.
- Increase the node frequency to try ILP heuristics.
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Multi index ILP neighbourhood

- With multi index variables $x_{i,j,k}$.
- Choose an index, for instance $j$.
- Given a subset $J^o$ of possible $j$, fix for all $(i, k)$ and $j_o \in J^o$ the variables $x_{i,j_o,k}$ to the current value.
- Huge diversity of possible neighbourhoods, some choices can be done using some knowledge of the problem.
RINS neighbourhoods

- Based on a linear relaxation of the problem: LP relaxation, LP relaxation with cuts of the root node, lagrangian relaxation . . .
- Fixing common integer values in a relaxation and the best solution known in a small ILP.
- Strengthening this neighbourhood: Fixing common integer values in several relaxations and the best solution known.

⇒ Non intuitive and generic neighbourhoods. Very small neighbourhoods, quick to compute.
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Unit Commitment Problem, UCP

UCP: Planning the production of electric power generating units, meeting the demands, in order to minimize the operations costs.

Cost function: Proportional cost to the power generated, fixed costs, start up costs.

Demand constraints on every time step:

- Demands in power. (forecasted)
- Demands in reserves, primary and secondary. (imposed)

Dynamic constraints:

- Min up- min down constraints. (start up)
- Minimum stops on discrete operating points. (in operation)

We model only discrete operating points.
Minimum stops constraints for fossil power plants
Minimum stops constraints for UCPd

\[ \Delta_u(i, i + 1) = \delta^{(i)} + \frac{P(u, i + 1) - P(u, i)}{|\text{grad}_{\text{montee}}(u)|}, \quad \Delta_u(i, i - 1) = \delta^{(i)} + \frac{P(u, i) - P(u, i - 1)}{|\text{grad}_{\text{baisse}}(u)|}. \]
Variable definition

- Level variables $l_{u,t}^{(i)}$: $l_{u,t}^{(i)} = 1$ iff unit $u$ generates at time step $t$ on a point $j \geq i$.

- Precedence constraints:
  $$\forall u, t, i, \quad l_{u,t}^{(i-1)} \leq l_{u,t}^{(i)}$$

- Efficient variables for branching and for ILP heuristics.
Efficient compact ILP formulation for UCPd

\[
\min \sum_{u,t} \left( C_{\text{start}}(u)y_{u,t'}^{(1)+} + C_{\text{fix}}(u)y_{u,t'}^{(1)} + \sum_{i \in \mathcal{I}_u^*} C_p(u)(P(u,i) - P(u,i-1))l_{u,t}^{(i)} \right)
\]

\[\forall u, t, i, l_{u,t}^{(i-1)} \leq l_{u,t}^{(i)}\] (1)

\[\forall t, \sum_{u \in U} \sum_{i \in \mathcal{I}_u} (P(u,i) - P(u,i-1))l_{u,t}^{(i)} \geq D_P(t)\] (2)

\[\forall j \in \{1, 2\}, \forall t, \sum_{u \in U} \sum_{i \in \mathcal{I}_u} (R_j(u,i) - R_j(u,i-1))l_{u,t}^{(i)} \geq D_{R_j}(t)\] (3)

\[\forall u, t, i, l_{u,t}^{(i+1)} \leq l_{u,t+1}^{(i)} \text{ and } l_{u,t}^{(i)} \geq l_{u,t+1}^{(i+1)}\] (4)

\[\forall u, t, i, y_{u,t}^{(i-1)} = y_{u,t}^{(i)+} + l_{u,t-1}^{(i)} - l_{u,t}^{(i)}\] (5)

\[\forall u, t, i, \sum_{t'=t+1}^{t+\Delta(i)+} y_{u,t'}^{(i+1)+} + \sum_{t'=t+1}^{t+\Delta(i)-} y_{u,t'}^{(i-1)-} \leq l_{u,t}^{(i)} - l_{u,t}^{(i+1)}\] (6)

\[\forall u, t, \sum_{t'=t-Lu+1}^{t} y_{u,t'}^{u} \leq l_{u,t}^{(1)} \text{ and } \sum_{t'=t-Lu+1}^{t} y_{u,t'}^{u} \leq l_{u,t-u}^{(1)}\] (7)
Plan

1. Facts and motivations

2. VNS scheme with ILP neighbourhoods
   - Algorithm
   - How to use an ILP solver in such a way?
   - Generic ILP neighbourhoods

3. Application to UCPd
   - Problem presentation and efficient ILP formulation
   - ILP neighbourhoods for UCPd

4. Results and perspectives
Specialized generic neighbourhoods

- Remove unused thermal units in the current solution. (selection in $u$)
- Fix the start up decisions. (selection in $i$ first level)
- Fix the decisions outside the time windows peaks (selection in $t$).
Time window fixing and propagated fixing

\[ l_{u,t}^{(i)} = 1 \]  \[ l_{u,t}^{(i)} = 0 \]  Direct fixing  Propagated fixing
Variable fixing when fixing mid power level

\[
\begin{align*}
\times & \quad l_{u,t}^{(i)} = 1 \\
\square & \quad l_{u,t}^{(i)} = 0 \\
\circ & \quad \text{Direct fixing} \\
\circ & \quad \text{Propagated fixing}
\end{align*}
\]
Neighbourhoods using special dynamic structures

Fixing based on coupling constraints (dynamic constraints for index $t$, and precedence constraint for index $i$):

- **1 translation**: fix variable in $t$ iff the correspondant value is the same at $t-1$ and $t+1$.

- Tube fixing: fix a variable in $i$ iff the correspondant value is the same at $i-1$ and $i+1$.

- More flexible decisions that could be crossed with other neighbourhoods.
Variable fixing with 1-translation

\[ l_{u,t}^{(i)} = 1 \]  
\[ l_{u,t}^{(i)} = 0 \]  
Direct fixing  
Propagated fixing
Tube fixing

- \( l_{u,t}^{(i)} = 1 \)
- \( l_{u,t}^{(i)} = 0 \)
- Direct fixing
- Propagated fixing
Mid level variable fixing with 1-translation

\[ l^{(i)}_{u,t} = 1 \quad \square \quad l^{(i)}_{u,t} = 0 \quad \odot \quad \text{Direct fixing} \quad \odot \quad \text{Propagated fixing} \]
### Implementation

- Sequential implementation, ILP solved with Cplex 12.3 with OPL, and OPL script language.
- Intel Core2 Duo processor, 2.80GHz.
- Maximal time resolution for the biggest neighbourhoods 30s, less than 10s for the easiest ILP.
### Results after 5 min time resolution

<table>
<thead>
<tr>
<th>Jeu de données</th>
<th>Meilleure Valeur à 5 min</th>
<th>%gap</th>
<th>heurSelect en 5 min</th>
<th>%gap</th>
<th>heurMixte en 5 min</th>
<th>%gap</th>
<th>VNS En 5 min</th>
<th>%gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataCharbon</td>
<td>16445498</td>
<td>0,93</td>
<td>16518267</td>
<td>0,44</td>
<td>16479774</td>
<td>0,21</td>
<td>16441401</td>
<td>-0,02</td>
</tr>
<tr>
<td>dataFuel</td>
<td>55911670</td>
<td>2,24</td>
<td>56924564</td>
<td>1,81</td>
<td>56799596</td>
<td>1,59</td>
<td>56791753</td>
<td>1,57</td>
</tr>
<tr>
<td>data48</td>
<td>34767116</td>
<td>3,23</td>
<td>35337301</td>
<td>1,64</td>
<td>35416050</td>
<td>1,87</td>
<td>35310761</td>
<td>1,56</td>
</tr>
<tr>
<td>data57</td>
<td>48940734</td>
<td>3,18</td>
<td>49803577</td>
<td>1,76</td>
<td>49789355</td>
<td>1,73</td>
<td>49795609</td>
<td>1,75</td>
</tr>
<tr>
<td>dataSpring</td>
<td>51087946</td>
<td>1,20</td>
<td>51377703</td>
<td>0,57</td>
<td>51215795</td>
<td>0,25</td>
<td>51212671</td>
<td>0,24</td>
</tr>
<tr>
<td>dataSpringWE</td>
<td>24427355</td>
<td>0,70</td>
<td>24587502</td>
<td>0,66</td>
<td>24594951</td>
<td>0,69</td>
<td>24612870</td>
<td>0,76</td>
</tr>
<tr>
<td>dataWinterWE</td>
<td>124481919</td>
<td>1,39</td>
<td>125426515</td>
<td>0,76</td>
<td>124898244</td>
<td>0,33</td>
<td>124845507</td>
<td>0,29</td>
</tr>
<tr>
<td>dataWinter</td>
<td>124194907</td>
<td>0,89</td>
<td>125278370</td>
<td>0,87</td>
<td>124392870</td>
<td>0,16</td>
<td>124434979</td>
<td>0,19</td>
</tr>
<tr>
<td>datSpring</td>
<td>17407479</td>
<td>0,42</td>
<td>17461794</td>
<td>0,31</td>
<td>17418736</td>
<td>0,06</td>
<td>17413560</td>
<td>0,03</td>
</tr>
<tr>
<td>datWinterWE</td>
<td>38837637</td>
<td>1,02</td>
<td>39344777</td>
<td>1,31</td>
<td>39088330</td>
<td>0,65</td>
<td>38920124</td>
<td>0,21</td>
</tr>
<tr>
<td>datWinter</td>
<td>34852129</td>
<td>0,50</td>
<td>35023821</td>
<td>0,49</td>
<td>34934800</td>
<td>0,24</td>
<td>34830070</td>
<td>-0,06</td>
</tr>
<tr>
<td>Total</td>
<td>520141776</td>
<td>1,36</td>
<td>525228623</td>
<td>0,98</td>
<td>523132677</td>
<td>0,58</td>
<td>522857143</td>
<td>0,52</td>
</tr>
</tbody>
</table>
General facts

- Initial question: are solution found by MIP solvers after a consequent resolution time local optima?

- Answer: NO, our VNS scheme can improve quickly solutions found after a long time with a frontal resolution.

- Solutions found in a few minutes can be better than the best fixing strategies after days.

- Efficiency of RINS heuristics, and its alternation with more intuitive neighbourhoods.
Advantages of the approach:

- Easy implementation once an ILP formulation is realized.
- No problem with infeasibilities: current solution stays always feasible. Well fitted for very constrained problems.
- Large number of neighbourhoods can be computed, with very different sizes. Very efficient to improve a local optimum.
- New neighbourhoods: RINS
Drawbacks of our approach:

- Efficiency relies on the ILP heuristics’ efficiency on the problem.

- Too many time in a neighbourhood search? Bad compromise calculus time/ improvement? Again, it depends on the problem.

- Needs to parametrize each ILP neighbourhood for a better efficiency (time limit . . . ).
Perspectives

- Parallel implementation. Using classical local search parallely?
- Application with more constraints.
- Application to another problem: the scheduling of nuclear power plants maintenances.
Thank you for your attention
Bibliography

- Dupin, N. and Bendotti, P.: ILP resolution of Unit Commitment problem with minimum stop constraints. in Proc COPI’11 - Conference on Optimization & Practices in Industry, Paris (2011)


- Cplex, reference manual