Column Generation and Branch and Price

Mads Kehlet Jepsen

Overview

Overview

- Column Generation for LP problems

Overview

- Column Generation for LP problems
- Column Generation example

Overview

- Column Generation for LP problems
- Column Generation example
- Branch and Price with branching on original variables

Overview

- Column Generation for LP problems
- Column Generation example
- Branch and Price with branching on original variables
- Branch-and-Price on Multi-Item Lot-sizing

Column generation idea in LP

Column generation idea in LP

- Identify a certain structure in the LP.

Column generation idea in LP

- Identify a certain structure in the LP.
- Split out the constraints that defines the structure as a subproblem.

Column generation idea in LP

- Identify a certain structure in the LP.
- Split out the constraints that defines the structure as a subproblem.
- Describe the polyhedron of the subproblem

Column generation idea in LP

- Identify a certain structure in the LP.
- Split out the constraints that defines the structure as a subproblem.
- Describe the polyhedron of the subproblem
- Solve the new linear optimization problem.

Some notation

- c is a vector of size n
- b is a vector of size m
- b' is a vector of size m'
- b'' is a vector of size m''
- A is a matrix of size $m \times n$
- A' is a matrix of size $m' \times n$
- A'' is a matrix of size $m'' \times n$
- x is the solution vector of size n

LP problem continued

A linear optimization problem is then stated as:

$$\begin{array}{lll} \max & cx & \\ \texttt{subjectto} & & \\ & A'x & = b' \\ & A''x & = b'' \\ & x & \geq 0 \end{array}$$

Polyhedral teory

Polyhedral teory

- A crossing of m half spaces is called a polyhedron

Polyhedral teory

- A crossing of m half spaces is called a polyhedron
- The equation $A''x = b$ forms a polyhedron.

Polyhedral teory

- A crossing of m half spaces is called a polyhedron
- The equation $A''x = b$ forms a polyhedron.

A polyhedron can be described by the M basic feasible solutions $v^1, v^2, ..., v^M$ and the N basic feasible directions $w^1, ..., w^N$.

Polyhedral teory

- A crossing of m half spaces is called a polyhedron
- The equation $A''x = b$ forms a polyhedron.

A polyhedron can be described by the M basic feasible solutions $v^1, v^2, ..., v^M$ and the N basic feasible directions $w^1, ..., w^N$.
The vector x satisfies $A''x = b$ if and only if:

$$x = \sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k$$

and $\sum_{k=1}^{M} r_k = 1$

Danzig-Wolfe Decomposition

Given:

$$\begin{array}{lll} \max & cx & \\ \texttt{subjectto} & & \\ & A'x & = b' \\ & A''x & = b'' \\ & x & \geq 0 \end{array}$$

Danzig-Wolfe Decomposition

Given:

$$\begin{array}{lll} \max & cx & \\ \texttt{subjectto} & & \\ & A'x & = b' \\ & A''x & = b'' \\ & x & \geq 0 \end{array}$$

Describe $A''x = b''$ using the feasible solutions and directions. Substitute this description into cx and $A'x$ original formulation.

Danzig-Wolfe Decomposition continued

We have:

$$x = \sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k$$

Danzig-Wolfe Decomposition continued

We have:

$$x = \sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k$$

Substituting into cx gives:

Danzig-Wolfe Decomposition continued

We have:

$$x = \sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k$$

Substituting into cx gives:

$$c(\sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k)$$

Danzig-Wolfe Decomposition continued

We have:

$$x = \sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k$$

Substituting into cx gives:

$$c(\sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k)$$

Substituting into $A'x$ gives:

$$A(\sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k)$$

Danzig-Wolfe Decomposition continued

$$
\begin{array}{lll}
\max & c(\sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k) & \\
\texttt{subjectto} & & \\
 & A'(\sum_{k=1}^{M} r_k v^k + \sum_{k=1}^{N} s_k w^k) & = b \quad (y') \\
 & \sum_{k=1}^{M} r_k & = 1 \quad (y'_{m'+1}) \\
 & r_k, s_k & \geq 0
\end{array}
$$

Delayed Column Generation

- Instead of enumerating all columns generate them when needed

Delayed Column Generation

- Instead of enumerating all columns generate them when needed
- Use the insight from the simplex algorithm and add a column when it has positive reduced cost.

Delayed Column Generation

- Instead of enumerating all columns generate them when needed
- Use the insight from the simplex algorithm and add a column when it has positive reduced cost.
- Given dual variable y' to $Ax = b$ and $y'_{m'+1}$ the reduced cost for a column is given as $\hat{c}_j = c_j - yA$

Delayed Column Generation

- Instead of enumerating all columns generate them when needed
- Use the insight from the simplex algorithm and add a column when it has positive reduced cost.
- Given dual variable y' to $Ax = b$ and $y'_{m'+1}$ the reduced cost for a column is given as $\hat{c}_j = c_j - yA$
- In our case we have $\hat{c}_k = c_k - y'(A'_k v^k)$

Delayed Column Generation Continued

Since any solution from the subproblem must satisfy $A''x = b''$ we obtain the following optimization problem called the pricing problem

Delayed Column Generation Continued

Since any solution from the subproblem must satisfy $A''x = b''$ we obtain the following optimization problem called the pricing problem

$$\begin{array}{lll} \max & \hat{c}x & \\ \texttt{subject to} & & \\ & A''x & = b'' \\ & x & \geq 0 \end{array}$$

$\hat{c} = c - y'(A')$

Pricing problems solution

If the subproblem is infeasible.

Pricing problems solution

If the subproblem is infeasible. The entire problem is infeasible.

Pricing problems solution

If the subproblem is infeasible. The entire problem is infeasible.

case i The subproblem has an optimal solution x^* such that $(c - y'A')x^* > y'_{m'+1}$

Pricing problems solution

If the subproblem is infeasible. The entire problem is infeasible.

case i The subproblem has an optimal solution x^* such that $(c - y'A')x^* > y'_{m'+1}$

case ii The subproblem has an optimal solution x^* such that $(c - y'A')x^* \leq y'_{m'+1}$

Pricing problems solution

If the subproblem is infeasible. The entire problem is infeasible.

case i The subproblem has an optimal solution x^* such that $(c - y'A')x^* > y'_{m'+1}$

case ii The subproblem has an optimal solution x^* such that $(c - y'A')x^* \leq y'_{m'+1}$

case iii The subproblem is unbounded

Pricing Problem case i and iii

The reduced cost of the column is positive and the entering column becomes :

$$\bar{a} = \begin{bmatrix} A'x \\ 1 \end{bmatrix}$$

Pricing Problem case i and iii

The reduced cost of the column is positive and the entering column becomes :

$$\bar{a} = \begin{bmatrix} A'x \\ 1 \end{bmatrix}$$

The reduced cost of the column is positive and the entering column becomes :

$$\bar{a} = \begin{bmatrix} A'x \\ 0 \end{bmatrix}$$

Initial solution and infeasibility

- Find some columns and that use these initially

Initial solution and infeasibility

- Find some columns and that use these initially
- Heuristic solutions from pricing problem can be used e.g solution that are not minimal

Initial solution and infeasibility

- Find some columns and that use these initially
- Heuristic solutions from pricing problem can be used e.g solution that are not minimal
- Initially the master problem may be infeasible

Initial solution and infeasibility

- Find some columns and that use these initially
- Heuristic solutions from pricing problem can be used e.g solution that are not minimal
- Initially the master problem may be infeasible
- Use the dual rays from initial infeasible master problem

Example from Chvatal chapter 26

$$A = \begin{bmatrix} 2 & 1 & -2 & -1 & 2 & -1 & -2 & -3 \\ 1 & -3 & 2 & 3 & -1 & 2 & 1 & 1 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 & -2 & -3 & 1 & 4 & 3 & -5 \end{bmatrix}$$

$$c = \begin{bmatrix} 9 & -1 & -4 & -2 & 8 & -2 & -8 & 12 \end{bmatrix}$$

Example Master problem

Devide A into:

$$A' = \begin{bmatrix} 2 & 1 & -2 & -1 & 2 & -1 & -2 & -3 \\ 1 & -3 & 2 & 3 & -1 & 2 & 1 & 1 \end{bmatrix}$$

$$b' = \begin{bmatrix} 4 & -2 & \end{bmatrix}$$

Pricing Problem

$$A'' = \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \end{bmatrix}$$

$$b'' = \begin{bmatrix} -3 & 1 & 4 & 3 & -5 \end{bmatrix}$$

$$c = \begin{bmatrix} 9 & -1 & -4 & -2 & 8 & -2 & -8 & 12 \end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \text{and} \qquad \tilde{A} = \begin{bmatrix} \quad \end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix} \quad \end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix}1\\1\\1\\0\\0\\0\\0\\0\end{bmatrix}, \qquad \text{and} \qquad \tilde{A} = \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix}4\end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$ and $$\tilde{A} = \begin{bmatrix} 1 & & & \\ 0 & & & \\ 0 & & & \end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix} 4 & & & \end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ and } \qquad \tilde{A} = \begin{bmatrix} 1 & 3 & \quad \\ 0 & -2 & \quad \\ 0 & 0 & \quad \end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix} 4 & 8 & \qquad \end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix}1\\1\\1\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}1\\1\\0\\0\\1\\0\\1\\0\end{bmatrix} \text{ and } \begin{bmatrix}3\\4\\0\\2\\0\\0\\0\\5\end{bmatrix} \quad \tilde{A} = \begin{bmatrix}1 & 3 & & \\ 0 & -2 & & \\ 0 & 0 & & \end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix}4 & 8 & & & \end{bmatrix}$$

Example initial master problem

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 3 \\ 4 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix} \quad \tilde{A} = \begin{bmatrix} 1 & 3 & -7 \\ 0 & -2 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{c} = \begin{bmatrix} 4 & 8 & -41 \end{bmatrix}$$

Eksample iteration 1

Solve:

$$\begin{array}{rrrrrr} \max & 4r_1 & +2r_2 & -41s_1 & & \\ \texttt{st} & & & & & \\ & 1s_1 & +3s_2 & -7r_1 & = & 4 \\ & & -2s_2 & 2r_1 & = & -2 \\ & & & 1r_1 & = & 1 \end{array}$$

Eksample iteration 1

Solve:

$$\begin{array}{rrrrcr} \max & 4r_1 & +2r_2 & -41s_1 & & \\ \texttt{st} & & & & & \\ & 1s_1 & +3s_2 & -7r_1 & = & 4 \\ & & -2s_2 & 2r_1 & = & -2 \\ & & & 1r_1 & = & 1 \end{array}$$

We obtain dual solution $y = [4, 2, -17]$

Eksample iteration 1

Solve:

$$
\begin{array}{rrrrrr}
\max & 4r_1 & +2r_2 & -41s_1 & & \\
\texttt{st} & & & & & \\
& 1s_1 & +3s_2 & -7r_1 & = & 4 \\
& & -2s_2 & 2r_1 & = & -2 \\
& & & 1r_1 & = & 1
\end{array}
$$

We obtain dual solution $y = [4, 2, -17]$

$\hat{c} = [-1, 1, 0, -4, 2, -2, -2, -2]$

Exsample itteration 1 continued

solve pricing problem:

$$
\begin{array}{lrrrrrrrrcr}
\max & -1 & 1 & 0 & -4 & 2 & -2 & -2 & -2 & & \\
 & -1 & & 1 & & 1 & & & & = & -3 \\
 & 1 & -1 & & 1 & & & & & = & 1 \\
 & & 1 & -1 & & & 1 & -1 & & = & 4 \\
 & & & & -1 & & -1 & & 1 & = & 3 \\
 & & & & & -1 & & 1 & -1 & = & -5 \\
 & x \geq 0 & & & & & & & & &
\end{array}
$$

Exsample itteration 1 continued

solve pricing problem:

$$\begin{array}{lrrrrrrrrcr}
\max & -1 & 1 & 0 & -4 & 2 & -2 & -2 & -2 & & \\
 & -1 & & 1 & & 1 & & & & = & -3 \\
 & 1 & -1 & & 1 & & & & & = & 1 \\
 & & 1 & -1 & & & 1 & -1 & & = & 4 \\
 & & & & -1 & & -1 & & 1 & = & 3 \\
 & & & & & -1 & & 1 & -1 & = & -5 \\
 & x \geq 0 & & & & & & & & &
\end{array}$$

The bounded solution becomes $x = [5, 4, 0, 0, 2, 0, 0, 3]$ with objective value -3

Example continued entering column

$$\begin{bmatrix} 2 & 1 & -2 & -1 & 2 & -1 & -2 & -3 \\ 1 & -3 & 2 & 3 & -1 & 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 9 \\ -6 \end{bmatrix}$$

Example continue objective

$$\begin{bmatrix} 9 & -1 & -4 & -2 & 8 & -2 & -8 & 12 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 3 \end{bmatrix} = 21$$

Conclusion on Column generation for LP

- Comparable with revised simplex

Conclusion on Column generation for LP

- Comparable with revised simplex
- Can be used when part of LP has special structure

Conclusion on Column generation for LP

- Comparable with revised simplex
- Can be used when part of LP has special structure
- Not much research in automatic decomposition.

Conclusion on Column generation for LP

- Comparable with revised simplex
- Can be used when part of LP has special structure
- Not much research in automatic decomposition.
- Some research in detecting subsets of model that could be solved using networkflow simplex.

Danzig-Wolfe decomposition for IP

$$
\begin{array}{lll}
\max & cx & \\
\texttt{subjectto} & & \\
 & A'x & = b \\
 & A''x & = d \\
 & x & \geq 0 \\
 & x \in \mathcal{N} &
\end{array}
$$

Master Problem

Assuming subproblem is bounded we obtain:

$$
\begin{array}{lll}
\max & c(\sum_{k=1}^{M} r_k v^k) & & \\
\texttt{subjectto} & & & \\
 & A'(\sum_{k=1}^{M} r_k v^k) & = b & (y') \\
 & \sum_{k=1}^{M} r_k & = 1 & (y'_{m'+1}) \\
 & r_k & \in \{0, 1\} &
\end{array}
$$

Pricing Problem

$\hat{c} = c - yA$

$$\begin{array}{lll} \max & \hat{c}x & \\ \texttt{subject to} & & \\ & A''x & = b'' \\ & x & \geq 0 \\ & x \in \mathcal{N} & \end{array}$$

Branch and Price

Branch and Price

- Relax the constraint $r_k \in \{0, 1\}$ to $0 \leq r_k \leq 1$

Branch and Price

- Relax the constraint $r_k \in \{0, 1\}$ to $0 \leq r_k \leq 1$
- Use delayed column generation to solve linear master problem.

Branch and Price

- Relax the constraint $r_k \in \{0, 1\}$ to $0 \leq r_k \leq 1$
- Use delayed column generation to solve linear master problem.
- Use different mip heuristics to obtain feasible solution on the lp relaxation.

Branch and Price

- Relax the constraint $r_k \in \{0, 1\}$ to $0 \leq r_k \leq 1$
- Use delayed column generation to solve linear master problem.
- Use different mip heuristics to obtain feasible solution on the lp relaxation.
- Impose branching decision to obatian optimality.

Branching

- Branching can be considered as a special case og cutting.

Branching

- Branching can be considered as a special case og cutting.
- If variable x_1 is fractional we obtain the two cuts:

Branching

- Branching can be considered as a special case og cutting.
- If variable x_1 is fractional we obtain the two cuts:

$$x_1 \leq \lfloor x_1 \rfloor x_1 \geq \lceil x_1 \rceil$$

Branching

- Branching can be considered as a special case og cutting.
- If variable x_1 is fractional we obtain the two cuts:

$$x_1 \leq \lfloor x_1 \rfloor x_1 \geq \lceil x_1 \rceil$$

In generel the cuts we obtain from branching are:

$$Fx \leq f$$

Branching on original variables

$$
\begin{array}{lll}
\max & cx & \\
\texttt{subjectto} & & \\
 & A'x & = b' \\
 & A''x & = b'' \\
 & x & \geq 0 \\
 & x \in \mathcal{N} &
\end{array}
$$

Branching on original variables

$$
\begin{array}{lll}
\max & cx & \\
\texttt{subjectto} & & \\
 & A'x & = b' \\
 & A''x & = b'' \\
 & x & \geq 0 \\
 & x \in \mathcal{N} & \\
 & Fx & \leq f
\end{array}
$$

Master Problem after branching

$$
\begin{array}{llll}
\max & c(\sum_{k=1}^{M} r_k v^k) & & \\
\texttt{subject to} & & & \\
 & A'(\sum_{k=1}^{M} r_k v^k) & = b & (y') \\
 & \sum_{k=1}^{M} r_k & = 1 & (y'_{m'+1}) \\
 & r_k & \in \{0, 1\} &
\end{array}
$$

Master Problem after branching

$$
\begin{array}{llll}
\max & c(\sum_{k=1}^{M} r_k v^k) & & \\
\texttt{subjectto} & & & \\
 & A'(\sum_{k=1}^{M} r_k v^k) & = b & (y') \\
 & \sum_{k=1}^{M} r_k & = 1 & (y'_{m'+1}) \\
 & r_k & \in \{0,1\} & \\
 & F(\sum_{k=1}^{M} r_k v^k) & \leq f & (\pi)
\end{array}
$$

Pricing Problem after branching

Reduced cost of a column becomes

$\hat{c}_k = c_k - y'(A'_k v^k) - \pi(F_k v^k)$

Pricing Problem after branching

Reduced cost of a column becomes

$\hat{c}_k = c_k - y'(A'_k v^k) - \pi(F_k v^k)$

$$\begin{array}{lll} \max & \hat{c}v & \\ \texttt{subject to} & & \\ & A''v & = b'' \\ & v & \geq 0 \end{array}$$

Multi-item Lot-Sizing

N items

T time periods

$d_{i,t}$ demand in period t for item i

$p_{i,t}$ production cost in period t for item i

$h_{i,t}$ holding cost in period t for item i

$f_{i,t}$ setup cost in period t for item i

$s_{i,t}$ stock in period t for item i

$x_{i,t}$ priduction in period t for item i

$y_{i,t}$ Indicates if machine is setup for item i in period t

Multi-item Lot-Sizing Model

A small bucket model where only one item can be produced in each time period.

$$
\begin{array}{llll}
\min & \sum_{i=1}^{N}\sum_{t=1}^{T} p_{i,t}x_{i,t} + h_{i,t}s_{i,t} + f_{i,t}y_{i,t} & & \\
& \sum_{i=1}^{N} y_{i,t} \quad \leq 1 & \texttt{for} & t = 1, ..., n \\
& s^{i}_{t-1} + x_{i,t} \quad = d_{i,t} + s_{i,t} & \texttt{for} & i = 1, ..., N, t = 1, ..., T \\
& x_{i,t} \quad \leq C_{i,t}y_{i,t} & \texttt{for} & i = 1, ..., N, t = 1, ..., T \\
& x_{i,t} \geq 0 & \texttt{for} & i = 1, ..., N, t = 1, ..., T \\
& s_{i,t} \geq 0 & \texttt{for} & i = 1, ..., N, t = 1, ..., T \\
& y_{i,t} \in \{0, 1\} & \texttt{for} & i = 1, ..., N, t = 1, ..., T
\end{array}
$$

Multi-item Lotsizing Master Problem

Multi-item Lotsizing Master Problem

- We have K_i production schdules for each item i.

Multi-item Lotsizing Master Problem

- We have K_i production schdules for each item i.
- The cost of a production schdule is $\tilde{c}_i^k = \sum_{t=1}^{T} p_{i,t}x_{i,t} + h_{i,t}s_{i,t}$

Multi-item Lotsizing Master Problem

- We have K_i production schdules for each item i.
- The cost of a production schdule is $\tilde{c}_i^k = \sum_{t=1}^{T} p_{i,t} x_{i,t} + h_{i,t} s_{i,t}$
- λ_i^k indicate if we choose the schedule

Multi-item Lotsizing Master Problem

- We have K_i production schdules for each item i.
- The cost of a production schdule is $\tilde{c}_i^k = \sum_{t=1}^{T} p_{i,t}x_{i,t} + h_{i,t}s_{i,t}$
- λ_i^k indicate if we choose the schedule

$$a_{i,t}^k =\leq \{ \begin{array}{ll} 1 & \texttt{if} \quad y_{i,t} = 1 \\ 0 & \texttt{otherwise} \end{array}$$

$$\begin{array}{lllll} \min & \tilde{c}\lambda_i^k & & & \\ & \sum_{i=1}^{N} \sum_{k=1}^{K_i} a_{i,t}^k \lambda_i^k & \leq 1 & \texttt{for} & t = 1,..,T \\ & \sum_{k=1}^{k_i} \lambda_i^k & = 1 & \texttt{for} & i = 1,..,N \\ & \lambda_i^k \in \{0,1\} & & & \end{array}$$

Exercises

You should solve Ex 1 and 2, if you have used alot time it is ok to skip 3 and 4.

Ex 1 Finish solving the example on the slides. Use the cplex files on the website.

Ex 2 State the mathematical model for the Multi-item lot-sizing pricing problem

Ex 3 Suggest a branching rule for the Multi-item lot-sizing Problem

Ex 4 Show how the branching rule influences the master problem.