Solving lexicographic multiobjective MIPs with Branch-Cut-Price

Marta Eso (The Hotchkiss School)
Laszlo Ladanyi (IBM T.J. Watson Research Center)
David Jensen (IBM T.J. Watson Research Center)
Outline

- Application: FCC Auction #31
- Solving IPs with branch-and-bound using an unusual relaxation
- Incorporating cutting planes to create branch-cut-price
- Treating the secondary objective via complementarity
- Application: FCC Auction #31 (revisited)
Wireless frequency licenses are auctioned off.

- **Iterative auction:** repeat until no more new bids
- **Bid submission:** regulated by complex rules (eligibility, bid survival, etc.) See Public Notices.
- **Bid evaluation:** given the bids, compute revenue-maximizing provisional allocation of licences.
- **Feedback:** bids’ standing, info on raising bid amounts for non-winning bids.

- Bids may be submitted for individual licences or for bundles of licences.
Application: FCC Auction #31 – 2

- primary objective: maximize revenue
- secondary objective: random (ensures random choice between alternate optima)
- Slow auction, with only a few rounds per day
- Target: bid evaluation and feedback computation in less than 15 mins
- These are IP’s that must be solved to optimality
- Major reservation against package bidding was its computational complexity
Bid evaluation

Stage I: Select a revenue-maximizing subset of bids
- consider bids from all rounds so far
- XOR of OR bids: bidder may win any bids from a round but all winning bids must come from the same round

Stage II: Select one of the optimal solutions randomly
- achieved by optimizing wrt random secondary objective
- traditionally implemented by adding the primary objective as a constraint
Feedback: provide price estimates for licences

- proper anonymous item prices do not exist in general unless bid evaluation can be solved as an LP
  - anonymous: price doesn’t depend on identity of bidder
  - proper: add up exactly to the bid amount for winning bids; to no less than the bid amount for non-winning bids

- price estimates: allow sum to drop below bid amount for non-winning bids
  - minimize the total drop, select solution that least deviates from prices in previous round
  - an LP followed by a QP – we solve the dual problems for numerical stability
Stage I: Revenue maximization

For each agent $a \in A$ and round $t \in T$ define:

- $M_{a,t}$: the matrix whose columns are the incidence vectors of bids
- $v_{a,t}$: the array of objective coeffs corresponding to these bids
- $x_{a,t}$: binary variables indicating which of these bids are accepted
- $y_{a,t}$: a binary variable indicating whether any of these bids are selected or not.
Stage I: disaggregated formulation

objective:
\[
\min \sum_{a,t} \begin{bmatrix} v_{a,t}^T, 0 \end{bmatrix} \begin{bmatrix} x_{a,t} \\ y_{a,t} \end{bmatrix}
\]

license constraints
\[
\sum_{a,t} \begin{bmatrix} M_{a,t}, 0 \end{bmatrix} \begin{bmatrix} x_{a,t} \\ y_{a,t} \end{bmatrix} \leq 1
\]

bidder constraints
\[
\sum_{t} \begin{bmatrix} 0^T, 1 \end{bmatrix} \begin{bmatrix} x_{a,t} \\ y_{a,t} \end{bmatrix} \leq 1 \quad \forall a
\]

bid-round constraints
\[
\sum_{a,t} \begin{bmatrix} M_{a,t}, -1 \end{bmatrix} \begin{bmatrix} x_{a,t} \\ y_{a,t} \end{bmatrix} \leq 0 \quad \forall a, t
\]

\[
x_{a,t}, y_{a,t} \in \{0, 1\} \quad \forall a, t
\]
Formulated by Dietrich & Forrest:

- Variables correspond to proposals: possible bid combinations of a bidder. The vector of variables for bidder $a$ is $\lambda_a$.

- Formulation of master problem
  - List proposals of bidder $a$ in $X_a$
  - Require that at most one proposal per bidder is selected
  - Require that $\lambda_a$'s are integral

- Subproblems used to dynamically generate proposals
Master Problem
\[ \min \sum_a v^T_a X_a \lambda_a \]
\[ \sum_a X_a \lambda_a \leq 1 \]
\[ e^T \lambda_a \leq 1 \quad \forall a \]
\[ \lambda_a \geq 0 \quad \forall a \]

Subproblems for each \( a, t \)
\[ \min (v^T_{a,t} - M^T_{a,t} \pi) x_{a,t} - \nu_a \]
\[ M_{a,t} x_{a,t} \leq 1 \]
\[ 1 \geq x_{a,t} \geq 0 \]
\[ x_{a,t} \text{ binary} \]

Solve via branch-and-bound.
In genIP problem considered

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{k} c_i^T x_i \\
\text{(IP)} & \quad \sum_{i=1}^{k} A_i x_i \leq b \\
& \quad D_i x_i \leq d_i \quad \forall i = 1, \ldots, k \\
& \quad x_i \text{ binary} \quad \forall i = 1, \ldots, k
\end{align*}
\]

- “hard” connecting constraints
- block-diagonal “easy” constraints
- binary requirement just for easier notation, trivial to relax to real MIP
Solving (IP) with Branch-and-Bound

Branching:
- any combination of changing bounds on constraints and/or variables (just to simplify discussion; easy to generalize)

Bounding:
- Instead of LP relaxation consider intersection of LP relaxation and the convex hull of IP solutions feasible to the easy constraints.
- Solve the bounding via Dantzig-Wolfe decomposition
Dantzig-Wolfe for bounding

Original relaxation (the \(b', d'_i\), and \(l_i, u_i\) vectors reflect the branching decisions)

\[
\begin{align*}
\min & \sum_{i=1}^{k} c_i^T x_i \\
\text{s.t.} & \sum_{i=1}^{k} A_i x_i \leq b' \\
& D_i x_i \leq d'_i \quad \forall i \\
& l_i \leq x_i \leq u_i \quad \forall i \\
& x_i \text{ integer} \quad \forall i
\end{align*}
\]

Dantzig-Wolfe decomposition:

Master Problem

\[
\begin{align*}
\min & \sum_{i=1}^{k} c_i^T X_i \lambda_i \\
\text{s.t.} & \sum_{i=1}^{k} A_i X_i \lambda_i \leq b' \\
& e^T \lambda_i = 1 \quad \forall i \\
& \lambda_i \geq 0 \quad \forall i
\end{align*}
\]

The \(i^{th}\) subproblem

\[
\begin{align*}
\min & (c_i^T - A_i^T \pi)x_i - \delta_i \\
\text{s.t.} & D_i x_i \leq d'_i \\
& l_i \leq x_i \leq u_i \\
& x_i \text{ integer}
\end{align*}
\]

\(\pi\): the dual vector corresponding to the “hard” constraints

\(\delta_i\): the dual value corresponding to the \(i^{th}\) convexity constraint.
Extension: cut generation

For a solution \((\lambda_1, \ldots, \lambda_k)\) to the Master Problem \((X_1\lambda_1, \ldots, X_k\lambda_k)\) is a solution to the original.

\[ \Rightarrow \text{generate cuts } \sum_{i=1}^{k} F_i x_i \leq f \text{ in the original space.} \]

\[ \Rightarrow \text{“Incorporate” } F \text{ into } A, \text{ i.e., add constraints} \]

\[ \sum_{i=1}^{k} F_i X_i \lambda_i \leq f \text{ to the Master Problem.} \]

The duals of the new constraints are incorporated into the objectives of the subproblems.

Note: there might be violated cuts for the master problem (in the traditional branch-and-price sense, i.e., when \(\lambda_i\) is assumed to be integer) while there are none for the original problem.
Extension: general branching, real MIP

- Extend branching from “change bounds” to “add cuts and change bounds”.
- Additional cuts are incorporated into $A$, the set of “hard” constraints.
- General bounds on the variables and allowing continuous variables trivially carry over to the subproblems.
End result: branch-cut-price

- Original formulation is never explicitly maintained
- \(\Rightarrow\) in effect branch-cut-price is implemented on the master problem where integrality of \(X_i \lambda_i\) is required

- In traditional branch-and-price integrality of \(\lambda_i\) is required
- \(\Rightarrow\) hence the trouble with cut generation (the duals of cuts generated for the master problem can’t be interpreted)
Lexicographic optimization

After optimizing wrt. a primary objective (Stage I.) we need to further optimize wrt. a secondary objective (Stage II.):

$$\min \sum_{i=1}^{k} v_i^T x_i$$

$$(IP - 2) \quad \sum_{i=1}^{k} A_i x_i \leq b$$

$$D_i x_i \leq d_i \quad \forall i = 1, ..., k$$

$$x_i \text{ binary} \quad \forall i = 1, ..., k$$

$x_i$ minimizes primary objective

Traditionally done by adding an extra constraint

$\Rightarrow$ degeneracy, numerical instability.
Stage II. solution method

- Explore Stage I. search tree.

- Discard leaves with lower bound > optimal primary value.

- In the rest of the leaves find alternate optimal solution with best secondary objective value and take best of those.
Evaluating a leaf

- Suppose all subproblems solved as LP when D-W terminated;
- \( \Rightarrow \) the leaf might as well have been bounded via LP relaxation;
- \( \Rightarrow \) can create dual optimal solution to original formulation;
- \( \Rightarrow \) can use complementarity to fix bounds to stay on LP optimal face;
- \( \Rightarrow \) primary objective will not change, can continue branch and bound with secondary objective.
Removing the "solve as LP" assumption

When D-W terminates, for each subproblem that does not solve as an LP do \textit{NOT} carry over the subproblem to Stage II, rather:

- Explore the search tree of the subproblem.
- Concentrate on the leaves where lower bound = optimal value
- For all such leaves
  - create a subproblem in Stage II. with the appropriate bound changes that define this leaf;
  - however, these subproblems will share the convexity constraint of the original subproblem.
Exploiting complementarity

Let $\pi$ be the dual vector in the master problem and $\gamma_i$’s be the dual vectors of the subproblems. Then $(\pi, \gamma_1, \ldots, \gamma_k)$ is dual optimal to the original formulation.

- if (in the original formulation) the reduced cost $c_i^j - \pi^T A_i^j - \gamma_i^T D_i^j$ of variable $x_i^j$ is negative (positive) then the variable must be fixed at its current upper (lower) bound for Stage II.

- if the dual value $\pi^k$ is negative (positive) then the $k^{th}$ row of the original problem (and the master problem) must be fixed at its current upper (lower) bound for Stage II.

- if the dual value $\gamma_i^k$ is negative (positive) then the $k^{th}$ row of the $i^{th}$ subproblem must be fixed at its current upper (lower) bound for Stage II.
FCC Auction #31: Stage I

objective: 
\[
\min \sum_{a,t} \left[ v_{a,t}^T, 0 \right] \left[ \begin{array}{c} x_{a,t} \\ y_{a,t} \end{array} \right]
\]

license constraints 
\[
\sum_{a,t} \left[ M_{a,t}, 0 \right] \left[ \begin{array}{c} x_{a,t} \\ y_{a,t} \end{array} \right] \leq 1
\]

bidder constraints 
\[
\sum_{t} \left[ 0^T, 1 \right] \left[ \begin{array}{c} x_{a,t} \\ y_{a,t} \end{array} \right] \leq 1 \quad \forall a
\]

bid-round constraints 
\[
\sum_{a,t} \left[ M_{a,t}, -1 \right] \left[ \begin{array}{c} x_{a,t} \\ y_{a,t} \end{array} \right] \leq 0 \quad \forall a, t
\]
\[
x_{a,t}, y_{a,t} \in \{0, 1\} \quad \forall a, t
\]

apply Branch-and-Bound to this formulation

bounding at search tree nodes is via Dantzig-Wolfe (bid-round + binary are "easy")
FCC Auction #31: Dantzig-Wolfe

Dantzig-Wolfe (license round constraints and $x$ binary are "easy"):

\[
\begin{align*}
\min & \sum_{a,t} \left[ v_{a,t}^T, 0 \right] \begin{bmatrix} X_{a,t} \\ y_{a,t} \end{bmatrix} \lambda_{a,t} \\
\sum_{a,t} [M_{a,t}, 0] \begin{bmatrix} X_{a,t} \\ y_{a,t} \end{bmatrix} \lambda_{a,t} & \leq 1 \\
\sum_{t} \left[ 0^T, 1 \right] \begin{bmatrix} X_{a,t} \\ y_{a,t} \end{bmatrix} \lambda_{a,t} & \leq 1 \quad \forall a \\
e^T \lambda_{a,t} & = 1 \\
\lambda_{a,t} & \geq 0
\end{align*}
\]

- replace = with $\leq$ in convexity constraints (0 is solution to subproblem)
Throughout column generation $\delta_{a,t}$ will always be 0 and $y_{a,t}$ will always be binary.

- $y_{a,t}$ will trivially be binary, no matter what $\delta_{a,t}$ is (since $x_{a,t}$ is binary).
- If all $y$’s are 0/1 then the bidder constraints dominate the convexity constraints hence there is an opt sol to the master problem with all $\delta$’s being 0.

In master problem discard convexity constraints (they’ll be always dominated by the bidder constraints)

In subproblems set $y$’s to 1 ($y_{a,t} = 0$ leads to a rather uninteresting problem).
FCC Auction #31: B-C-P formulation

Master Problem
\[
\begin{align*}
\text{min } & \sum_{a,t} v_{a,t}^T X_{a,t} \lambda_{a,t} \\
\text{subject to } & \sum_{a,t} M_{a,t} X_{a,t} \lambda_{a,t} \leq 1 \\
& \sum_{t} e^T \lambda_{a,t} \leq 1 \quad \forall a \\
& \lambda_{a,t} \geq 0 \quad \forall a, t
\end{align*}
\]

Subproblems
\[
\begin{align*}
\text{min } & \left( v_{a,t}^T, -M_{a,t}^T \pi \right) x_{a,t} - \nu_a \\
\text{subject to } & M_{a,t} x_{a,t} \leq 1 \\
& 1 \geq x_{a,t} \geq 0 \\
& x_{a,t} \text{ binary}
\end{align*}
\]

Note: was non-trivial to eliminate the \( y \) variables.
Flashback: Dietrich and Forrest

- proposals: possible bid combinations of a bidder.

Formulation of master problem
- List proposals of bidder $i$ in $X_i$
- Require that at most one proposal per bidder is selected
- Require that $\lambda_i$'s are integral

Subproblems used to dynamically generate proposals

Solution method is branch-and-price; lower bounding through LP relaxation.
Flashback: Dietrich and Forrest - 2

- D&F lower bounding problem is *identical* to ours.
- D&F branching rule must be based on naive formulation to be able to do branch-and-price (consistency)

⇒ D&F search tree and our search tree uses same bounding and same branching: same formulation!

⇒ intuitive column generation is the same as Dantzig-Wolfe based.
branching on license: whether or not a license is assigned to a particular bidder. The “yes” side is enforced by

- setting $= 1$ the constraint of this licence in all the subproblems of this bidder
- eliminating conflicting bids from other bidders’ subproblems
- setting $= 1$ the constraint of this licence in MP

generated clique and odd hole inequalities
Stage I. computation is fast (the subproblems usually solve as LPs)

Stage II. is instantaneous, in effect the problem is fixed. Compare this to the natural approach, where:
- matrix contains a fully dense row (primary obj)
- matrix ill-conditioned (mostly 0-1, coeffs in new row $10^7$)
- objective coeffs also $10^8$
- ⇒ dangerously close to machine precision ($10^{-17}$)

Implementation used the BCP framework and the Cut Generation Library from http://www.coin-or.org
**Tips and tricks (a.k.a. enhancements)**

- Preprocessing: eliminate rounds and bids that cannot be winners (dominated by bids from a later round by the same bidder)
- Enumeration of small subproblems into proposals.
- Presolving subproblems at the beginning of each search tree node
- FCC bids (reserve prices are implemented by a single bid for each license): instead of creating a subproblem for the FCC, subtract the reserve amounts from all other bids.
Data and results

- Current plan calls for auctioning 12 licences, could be solved by enumeration of all proposals (see Dietrich and Forrest).

- Problems from the FCC’s problem generator.
  - 12 licences, up to 44 rounds, 6-7000 bids, up to 30 bidders. 60-80 percent removed by preprocessing (we have 20-30 instances of this) under 2 seconds
  - 50 licences, 15K bids, 16 rounds, 50 bidders (we have 5 instances) about 2.5 minutes; second stage never takes more than a couple of seconds – this is usually the difficult stage. stages 3-4 less than 15 seconds.
  - 150 licences, 10K bids, 50 bidders, 4 rounds (1 instance) about 20 minutes; second stage no more than a couple of seconds. stages 3-4 less than a minute
Conclusion and future work

Using branch-cut-price was very effective for this application. We plan to experiment with solving general MIPs:

- Find a set of separator rows
- Treat the separator as the “hard” constraint
- Treat the decomposed small problems individually as the subproblems.