Branch and bound for biobjective mixed integer optimization

Vienna, 12 September 2014

Pietro Belotti
Xpress Optimization
Birmingham, UK

Banu Soylu
Erciyes University
Kayseri, Turkey

Margaret Wiecek
Clemson University
Clemson, USA

Biobjective MILP

$$\begin{aligned} \min \quad & y_1 = \boldsymbol{c}_1^\top \boldsymbol{x} \\ \min \quad & y_2 = \boldsymbol{c}_2^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x} \in X \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p}) \end{aligned}$$

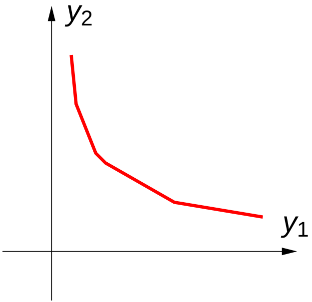
where $X = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$. Assume X is bounded.

Pareto set of a biobjective **linear** problem

Consider the problem $\min\{(\boldsymbol{c}_1^\top \boldsymbol{x}, \boldsymbol{c}_2^\top \boldsymbol{x}) : A\boldsymbol{x} \leq \boldsymbol{b}\}$.

Y is a union of segments.

The set $Y + \mathbb{R}^2_+$ is convex.



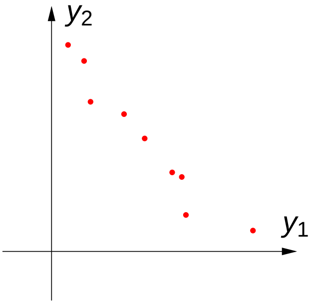
FICO

Pareto set of a biobjective **pure integer** problem

Consider the problem $\min\{(\boldsymbol{c}_1^\top \boldsymbol{x}, \boldsymbol{c}_2^\top \boldsymbol{x}) : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^n\}$.

Y is finite.

The set $Y + \mathbb{R}^2_+$ is nonconvex.



FICO

Pareto set of a biobjective **pure integer** problem

- We can generate all (non-dominated) solutions by exhaustive search in finite time.
- The Pareto set is given by a subset of the leaf nodes of the branch-and-bound tree.
- Same for the **mixed integer** case if $\boldsymbol{c}_1$ or $\boldsymbol{c}_2$ are nonzero only in the integer components

Pareto set of a biobjective **mixed integer** problem

- Consider $\bar{X}$ as the set of all *fixings* of the integer variables $\boldsymbol{x}_{1:p} = \bar{\boldsymbol{x}}$:

$$\bar{\boldsymbol{x}} \in \bar{X} \subset \mathbb{Z}^p$$

and $\bar{\boldsymbol{x}}$ is such that $\{\boldsymbol{x} \in X : \boldsymbol{x}_{1:p} = \bar{\boldsymbol{x}}\} \neq \emptyset$.

$\Rightarrow$ We get an LP Pareto set $Y_{\bar{\boldsymbol{x}}}$ for each $\bar{\boldsymbol{x}} \in \bar{X}$

- Y is a subset of the union of all these Pareto sets:

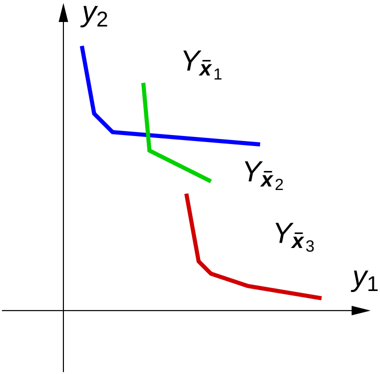
$$Y \subseteq \bigcup_{\bar{\boldsymbol{x}} \in \bar{X}} Y_{\bar{\boldsymbol{x}}}$$

(just eliminate its dominated points)

FICO

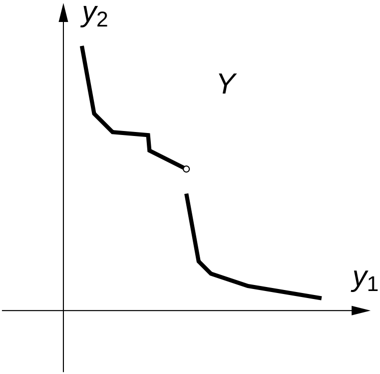
Example

Suppose $|\bar{X}| = 3$, i.e., $\bar{X} = \{\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \bar{\boldsymbol{x}}_3\}$.



Example

Suppose $|\bar{X}| = 3$, i.e., $\bar{X} = \{\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2, \bar{\boldsymbol{x}}_3\}$.



Previous work

- Fathoming rules: Mavrotas & Diakoulakis '98; Ehrgott & Gandibleux '08, Delort & Spanjaard '10
- TSP: BB where every node is a polynomially solvable BO-ILP (Jozefoviez, Laporte, Semet '11)
- BOMILP with one pure integer objective function: Costa, Captivo, Climaco '08; Stidsen, Andersen, Dammann '12; Prins, Prodhon, Wolfler Calvo '06
- Partitioning the objective space and solving multiple MILPS (Savelsbergh & Boland '13)
- Bi-objective MINLP (D'Ambrosio and Cacchiani, previous talk)

Branch-and-bound algorithms for BO-MIPS

Main idea: **one** run of the branch-and-bound.

- BB methods for MILP are extremely sophisticated and flexible
- BB is a tree search with rules to drop entire subtrees
- Relaxations are refined after branching
- BB gives estimate how close we are to termination (gap)

FICO

Branch-and-bound scheme

procedure BB $(\boldsymbol{c}_1, \boldsymbol{c}_2, A, \boldsymbol{b})$

$\mathcal{L} \leftarrow \{P\}$

while $\mathcal{L} \neq \emptyset$ **do**

Select node P_k from $\mathcal{L}$

if P_k cannot be fathomed **then**

Solve (P_k)

if P_k not fully searched **then**

Branch P_k into P'_k and P''_k

$\mathcal{L} \leftarrow (\mathcal{L} \setminus \{P_k\}) \cup \{P'_k, P''_k\}$

Solving BO-MIPS

Generalizing branch-and-bound from MIPS to BO-MIPS requires:

- a node solver (no longer an LP $\rightarrow$ parametric simplex method)
- fathoming rules (no longer “`lower_bound >= cutoff`”)
- termination criteria (redefine the gap)

Fathoming rules for node k

Rule 1: Fathom if LP relaxation infeasible;

Rule 2: Fathom if its Pareto set Y_k is dominated by the Pareto set Y of the original problem.

However, at node k we know neither Y_k nor Y.

Fathoming rules

Instead of Y:

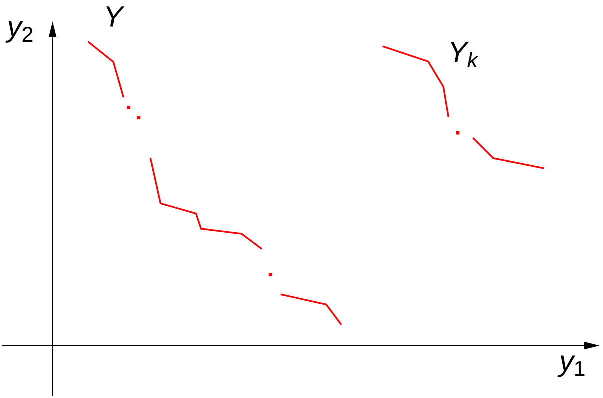
- take a set Y^{cutoff} of **feasible** solutions that are not dominated by others encountered so far
- Y^{cutoff} can be updated by adding any integer feasible $\boldsymbol{x}$

Instead of Y_k:

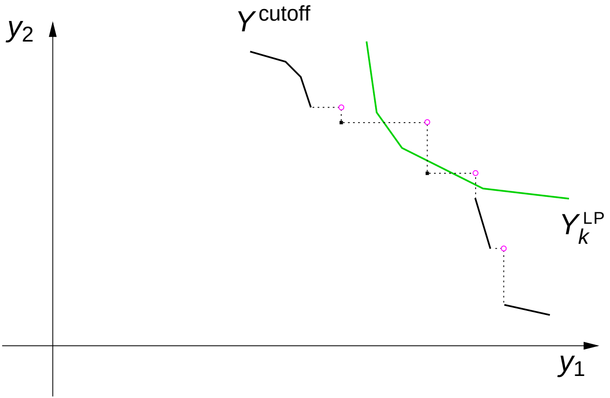
- Consider the Pareto set Y_k^{LP} of the LP relaxation of the BO-MIP at node k

Then a sufficient condition for fathoming k is that each point $\boldsymbol{y}$ of Y_k^{LP} is dominated by at least one point of Y^{cutoff}.

7 pictures are worth 70 words

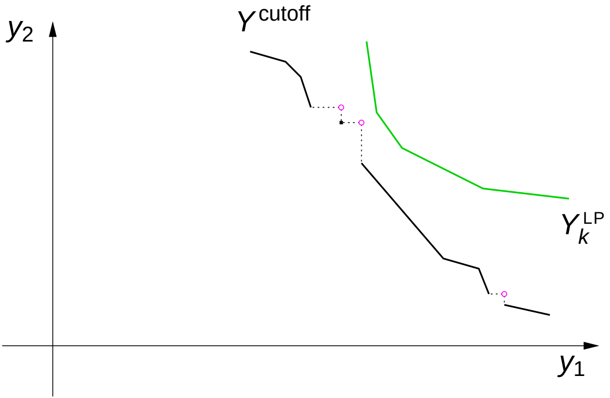


7 pictures are worth 70 words



FICO

7 pictures are worth 70 words



FICO

Nadir points and segments

- Every $\circ$ in the picture is a nadir point.
- We use nadir points and *segments* to create a sufficient condition for fathoming:

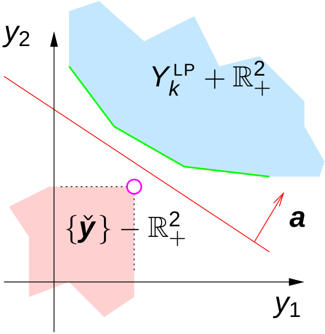
procedure FATHOM $(Y_k^{\mathrm{LP}}, Y^{cutoff})$
  `fathom` $\leftarrow$ true
  for each nadir point $\check{\boldsymbol{y}}$ **do**
    Find line separating $Y_k^{\mathrm{LP}} + \mathbb{R}^2_+$ from $\{\check{\boldsymbol{y}}\} - \mathbb{R}^2_+$
    if no such line exists **then**
      `fathom` $\leftarrow$ false
      break
  for each segment $(\boldsymbol{y}', \boldsymbol{y}'')$ **do**
    $[\ldots]$
  if `fathom` $=$ true **then**
    fathom node

Fathoming rule: nadir points $\check{\boldsymbol{y}}$

The separating line has coefficient vector $\boldsymbol{a} = (\lambda, 1 - \lambda)$.
Hence we seek $\lambda \in [0, 1]$ such that

$$\lambda \check{y}_1 + (1 - \lambda)\check{y}_2 \leq \lambda y_1 + (1 - \lambda) y_2$$

for every $\boldsymbol{y} = (y_1, y_2) \in Y_k^{\mathrm{LP}}$



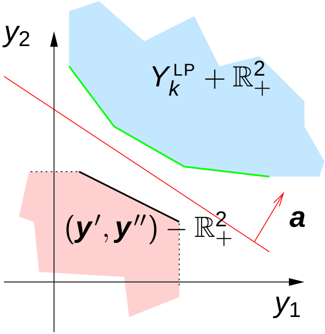
FICO

Fathoming rule: segments $(\boldsymbol{y}', \boldsymbol{y}'')$

The separating line has coefficient vector $\boldsymbol{a} = (\lambda, 1-\lambda)$.
Hence we seek $\lambda \in [0,1]$ such that

$$\lambda y_1' + (1-\lambda) y_2' \leq \lambda y_1 + (1-\lambda) y_2$$
$$\lambda y_1'' + (1-\lambda) y_2'' \leq \lambda y_1 + (1-\lambda) y_2$$

for every $\boldsymbol{y} = (y_1, y_2) \in Y_k^{\text{LP}}$



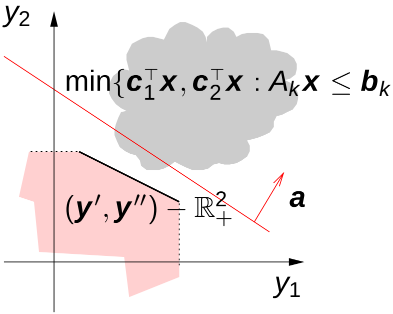
Fathoming rule: segments $(\boldsymbol{y}', \boldsymbol{y}'')$

The separating line has coefficient vector $\boldsymbol{a} = (\lambda, 1 - \lambda)$.
Hence we seek $\lambda \in [0, 1]$ such that

$$\lambda y_1' + (1 - \lambda) y_2' \leq \lambda y_1 + (1 - \lambda) y_2$$
$$\lambda y_1'' + (1 - \lambda) y_2'' \leq \lambda y_1 + (1 - \lambda) y_2$$

for every $\boldsymbol{y} = (y_1, y_2) \in Y_k^{\text{LP}}$

Wait . . . we don't have Y_k^{LP}



Fathoming rule: model (nadir points only)

In general: can separate nadir point $\check{\boldsymbol{y}}$ if

$$\lambda\check{y}_1 + (1-\lambda)\check{y}_2 \leq \lambda \boldsymbol{c}_1^\top \boldsymbol{x} + (1-\lambda)\boldsymbol{c}_2^\top \boldsymbol{x} \qquad \forall\, \boldsymbol{x} \in X_k^{\mathsf{LP}}$$

that is, if we can find a $\lambda \in [0,1]$ such that

$$\begin{aligned}
\lambda\check{y}_1 + (1-\lambda)\check{y}_2 &\leq \min\{\lambda \boldsymbol{c}_1^\top \boldsymbol{x} + (1-\lambda)\boldsymbol{c}_2^\top \boldsymbol{x} : \boldsymbol{x} \in X_k^{\mathsf{LP}}\} \\
&= \min\{\lambda \boldsymbol{c}_1^\top \boldsymbol{x} + (1-\lambda)\boldsymbol{c}_2^\top \boldsymbol{x} : A_k \boldsymbol{x} \leq \boldsymbol{b}_k\} \\
&= \max\{\boldsymbol{b}_k^\top \boldsymbol{w} : A_k^\top \boldsymbol{w} = \lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2, \boldsymbol{w} \leq \boldsymbol{0}\}
\end{aligned}$$

Fathoming rule: model (nadir points only), cont'd

Hence, if we find $\boldsymbol{w}, \lambda$ such that

$$\begin{aligned} &\lambda \check{y}_1 + (1-\lambda)\check{y}_2 \leq \boldsymbol{b}_k^\top \boldsymbol{w} \\ &A_k^\top \boldsymbol{w} = \lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2 \\ &\boldsymbol{w} \leq \boldsymbol{0} \\ &\lambda \in [0,1] \end{aligned}$$

we can separate $\check{\boldsymbol{y}}$ from $Y_k^{\mathrm{LP}} + \mathbb{R}_+^2$

Fathoming rule: separation of nadir points

$$\begin{aligned} \min \quad & \lambda \check{y}_1 + (1-\lambda)\check{y}_2 - \boldsymbol{b}_k^\top \boldsymbol{w} \\ \text{s.t.} \quad & A_k^\top \boldsymbol{w} = \lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2 \\ & \boldsymbol{w} \leq \boldsymbol{0} \\ & \lambda \in [0,1] \end{aligned}$$

If optimal solution has positive objective function, we cannot fathom k. Otherwise, check the next nadir point.

- In theory, must solve an LP for every nadir point $\check{\boldsymbol{y}}$
- In practice, we can limit the actual number of LPs solved
- For segments, we need an extra constraint

Fathoming rule: separation of nadir segment $[\boldsymbol{y}', \boldsymbol{y}'']$

$$
\begin{array}{ll}
\min & \lambda \check{y}_1' + (1-\lambda)\check{y}_2' - \boldsymbol{b}_k^\top \boldsymbol{w} \\
\text{s.t.} & \lambda \check{y}_1'' + (1-\lambda)\check{y}_2'' \leq \boldsymbol{b}_k^\top \boldsymbol{w} \\
& A_k^\top \boldsymbol{w} = \lambda \boldsymbol{c}_1 + (1-\lambda)\boldsymbol{c}_2 \\
& \boldsymbol{w} \leq \boldsymbol{0} \\
& \lambda \in [0, 1]
\end{array}
$$

If an optimal solution has nonpositive objective function value, proceed to the next nadir point/segment. If the problem is infeasible or has a positive optimal objective, the node cannot be fathomed.

Fathoming rule: one last simplification

Fathoming LP:

$$\begin{array}{rlcll} -\check{y}_2 + \max & \boldsymbol{b}_k^\top \boldsymbol{w} + (\check{y}_2 - \check{y}_1)\lambda & & & \\ \text{s.t.} & A_k^\top \boldsymbol{w} + (\boldsymbol{c}_2 - \boldsymbol{c}_1)\lambda & = & \boldsymbol{c}_2 & (\boldsymbol{x}) \\ & \lambda & \leq & 1 & (\mu) \\ & \boldsymbol{w} \leq \boldsymbol{0}, \lambda \geq 0. & & & \end{array}$$

Its dual is somewhat familiar:

$$\begin{array}{rl} -\check{y}_2 + \min & \boldsymbol{c}_2^\top \boldsymbol{x} + \mu \\ \text{s.t.} & A_k \boldsymbol{x} \leq \boldsymbol{b}_k \\ & (\boldsymbol{c}_2 - \boldsymbol{c}_1)^\top \boldsymbol{x} + \mu \geq (\check{y}_2 - \check{y}_1) \\ & \mu \geq 0 \end{array}$$

It's the original node problem amended with one variable and one constraint (only slightly more involved for segments).

FICO

Speeding up fathoming

Large problems $\Rightarrow$ many nadir points available at every node.

- Restrict to nadir pts. dominated by ideal point of node LP
- The solution $(\boldsymbol{x}^\star, \mu^\star)$ to one fathoming LP

$$
\begin{array}{rl}
-\check{y}_2 + \min & \boldsymbol{c}_2^\top \boldsymbol{x} + \mu \\
\text{s.t.} & A_k \boldsymbol{x} \leq \boldsymbol{b}_k \\
& (\boldsymbol{c}_2 - \boldsymbol{c}_1)^\top \boldsymbol{x} + \mu \geq (\check{y}_2 - \check{y}_1) \\
& \mu \geq 0
\end{array}
$$

can be used to exclude any other nadir point $\bar{\boldsymbol{y}}$ such that

$$
\begin{array}{c}
-\bar{y}_2 + \boldsymbol{c}_2^\top \boldsymbol{x}^\star + \mu^\star \leq 0 \\
(\boldsymbol{c}_2 - \boldsymbol{c}_1)^\top \boldsymbol{x}^\star + \mu^\star \geq (\bar{y}_2 - \bar{y}_1)
\end{array}
$$

- If condition not met for $\bar{\boldsymbol{y}}$, try a more suitable $\mu \neq \mu^\star$.

FICO

Implementation

Used a commercial MILP solver. Added callbacks for

- **node solving**: used the parametric simplex algorithm (PSA)
- **branching**: branching variable found during PSA
- **presolve**: for fathoming rules

Also, disabled cuts and presolving that use objective information (the solver must be oblivious of the two objectives).

A few experiments

var	con	$\mathbb{R}$	0-1	int	CPU(s)	nodes	infeas	fathom	PSA
20	20	10	5	5	0.62	78.0	22.0	6.0	52.8
20	20	10	10	0	0.54	70.0	21.0	5.4	46.2
20	20	10	0	10	0.56	66.0	21.4	5.6	41.4
40	40	20	10	10	3.32	238.0	67.6	34.6	136.8
40	40	20	20	0	2.38	164.4	38.2	30.0	97.0
40	40	20	0	20	4.00	281.6	61.6	55.0	162.2
60	60	30	15	15	10.86	458.4	108.4	74.6	251.2
60	60	30	30	0	14.38	638.4	171.0	89.8	338.2
60	60	30	0	30	15.90	738.8	212.4	70.4	392.8
80	80	40	20	20	22.34	616.8	156.0	71.2	328.4
80	80	40	40	0	37.58	1026.8	203.6	165.6	538.2
80	80	40	0	40	53.12	1414.8	279.4	219.2	738.4

The alternative

Compare with the ϵ-constraint method:

procedure ϵ-CONSTRAINT $(\boldsymbol{c}_1, \boldsymbol{c}_2, A, \boldsymbol{b})$

  for $\epsilon \in \{\epsilon_1, \epsilon_2, \ldots, \epsilon_q\}$ **do**

    Solve $\min\{\boldsymbol{c}_1^\top \boldsymbol{x} : \boldsymbol{c}_2^\top \boldsymbol{x} \leq \epsilon, \boldsymbol{x} \in X \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})\}$

    Save optimal solution $\boldsymbol{x}^\star$ as a Pareto point

Problems:

- It only works on discrete Pareto sets
- It is $\approx 20\times$ slower
- (it requires q single-objective MIPs $\ldots$)

Current research

Storing a large number of Pareto points (and the corresponding solutions) requires a data structure with

- fast insertion (with elimination of dominated points)
- fast check for dominance

We are integrating the branch-and-bound implementation above with a data structure that can be used for fast insertion of new points and that holds the Pareto set at the end of the BB[1].

[1]Adelgren, B., Gupte, Efficient storage of Pareto points in bi-objective mixed integer programming, in preparation.

References

P. Belotti, B. Soylu, M.M. Wiecek, “A branch-and-bound algorithm for biobjective mixed-integer programs”.

http://www.optimization-online.org/DB_HTML/2013/01/3719.html

Thank You

Pietro Belotti
pietrobelotti@fico.com