Multi-criteria linear programming with reverse search

Sebastian Schenker

TU Berlin / Zuse Institute Berlin

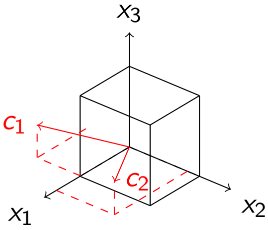
12th September 2014

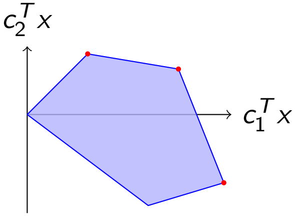
Outline

Multi-criteria linear problem

- Given $C \in \mathbb{R}^k$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

$$\begin{aligned} \min / \max\, & Cx \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$



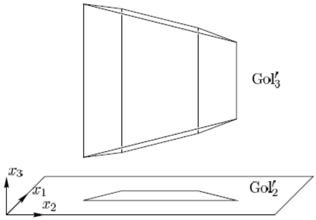


Preliminaries

- Basis B is set of m column indices of A with corresponding regular matrix $\left(A_{B(1)}A_{B(2)}\dots A_{B(m)}\right)$
- Co-basis N set of non-basic indices
- Pivoting to adjacent basic feasible solution by pivoting non-basic index $j \in N$ into B
- $x^* \in \mathcal{X}$ efficient solution if and only if there exists $\lambda \in \mathbb{R}^k_+$ such that x^* is optimal solution to $\min / \max \lambda^T Cx$ s.t. $x \in \mathcal{X}$

Bi-criteria linear program with 2^{n-1} non-dominated vertices based on deformed cube

$$
\begin{gathered}
\max\left(x_{n-1}\,,\,x_n\right)\\
\text{s.t. } 0 \leq x_1 \leq 1,\\
\epsilon x_1 \leq x_2 \leq 1-\epsilon x_1,\\
\epsilon(x_i-\gamma x_{i-1}) \leq x_{i+1} \leq 1-\epsilon(x_i-\gamma x_{i-1})\\
\text{for } 2 \leq i \leq n,\ \epsilon < 0.5,\ \gamma \leq 0.25\epsilon.
\end{gathered}
$$



Source: "Deformed products and maximal shadows of polytopes" by Amenta, Ziegler

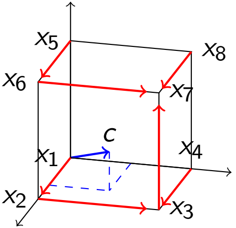
Conventional graph searches

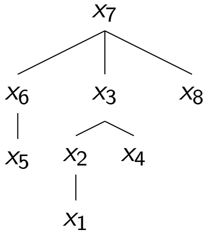
```
initialize data structure S                    ▷ Depth-first-search
while S is not empty do
    v ← S.pop();
    if v is not labeled yet then
        label v;
        for all w adjacent to v do
            S.push(w)
        end for
    end if
end while
```

- Graph searches use datastructure (e.g. stack, priority queue) to keep track of already visited nodes
- Number of elements in data structure might be huge

Reverse search

- Coined by Avis and Fukuda: “Reverse search for enumeration”
- Enumeration of vertices of polyhedra without book-keeping
- Consider paths generated by simplex from any vertex towards optimal vertex
- Paths yield a tree that can be traversed



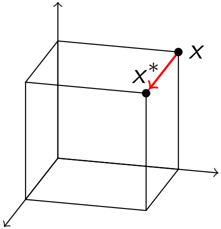


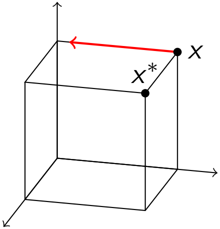
Traversing the reverse search tree

- Let x^* be currently considered vertex with corresponding basis B^*
- Let $N = \{N(1), \ldots, N(n-m)\}$ be set of non-basic indices
- Simulate pivoting $N(1)$ into B^* yielding B
- Check whether pivot rule w.r.t. objective c pivots from B to B^*

  if yes: pivot $N(1)$ into B^*

  if no: neglect $N(1)$ and check for $N(2)$
- apply recursively



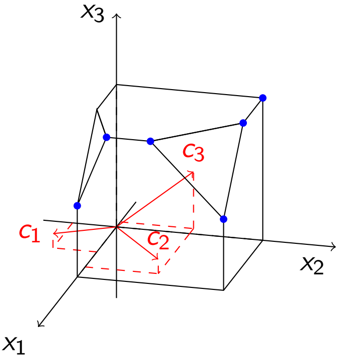


Pseudo-code for reverse search

```
B ← B*, N ← N*, j ← 1;
repeat
    while j ≤ |N*| do
        v ← N(j);
        if reverse(B, v, u) then;
            pivot(B, v, u);
            if lexmin(B,0) then
                output current vertex;
            end if
            j = 1;
        else
            j ← j + 1;
        end if
    end while
    selectpivot(B, r, j);
    pivot(B, r, N(j));
    j ← j + 1;
until j > |N*| ∧ B == B*
```

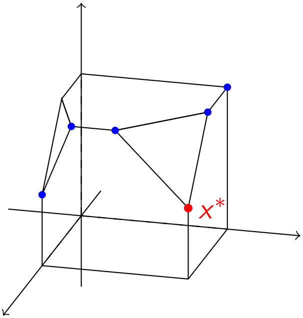
Enumerate efficient solutions without book-keeping

- Efficient solutions are connected
- Goal: Use reverse search approach for enumerating efficient solutions



Phase 1

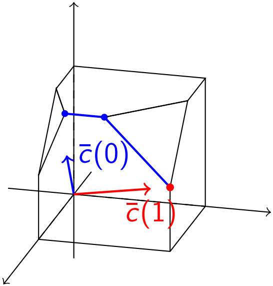
- Compute efficient solution x^* that acts as root node



- Which scalarisation of $c_1, \ldots, c_k$ shall be used as objective for reverse search?
- Problem: arbitrary but fixed (scalarized) objective might connect efficient solution with non-efficient solution

The path of the parametric simplex

- Let x^* be efficient solution computed in Phase 1
- Let c^* be objective which optimizes x^*
- Let x be another efficient solution
- Let c be objective which optimizes x
- Consider $\bar{c}(\lambda) = \lambda c^* + (1 - \lambda)c$ for $\lambda \in [0, 1]$



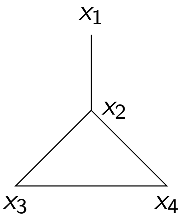
- Path taken by $\bar{c}(\lambda)$ connects only efficient solutions with x^* as terminal solution

Idea: Change considered objective after each step

- Consider $\bar{c}(\lambda) = \lambda c^* + (1 - \lambda)c$ until first break-point
  - If $x \neq x^*$, then x is optimal for $\bar{c}(\lambda)$ for $\lambda \in [0, \bar{\lambda}]$ with $\bar{\lambda} < 1$
  - Some $\hat{x}$ becomes optimal for $\bar{c}(\lambda)$ for $\lambda > \bar{\lambda}$
  - Let $\hat{c}$ be objective for which $\hat{x}$ is optimal
- Change considered objective to $\lambda c^* + (1 - \lambda)\hat{c}$
- x^* remains terminal solution for $\lambda = 1$
- In each step parametric objective can be computed from tableau of current basic feasible solution and root tableau

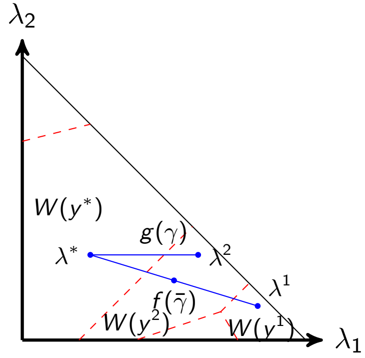
Correctness and completeness of the algorithm

- Correctness: Algorithm visits only efficient solutions. ✓
- Completeness: Can cycles occur?



Completeness of the algorithm

- Weight space: $W(\bar{y}) = \{\lambda \in \mathbb{R}^k_+ : \lambda^T \bar{y} \leq \lambda^T y \text{ for } y \in \mathcal{Y}\}$
- $W(\bar{y})$ convex
- Apply separation theorem for convex sets



$$\lambda \in \mathbb{R}^3_+, \lambda_1 + \lambda_2 + \lambda_3 = 1$$

Thanks for your attention.