Similarity-based Lexicographic Method for Hierarchical Multiobjective Linear Programs

Devanand R, Ashutosh Mahajan and N. Hemachandra

Department
Industrial Engineering and Operations Research,
IIT Bombay, India

Acknowledgment: *I thank Tushar Shekhar, my external advisor, and manager at Blue Yonder, for his support and suggestions.*

At High Level

We define the **concept of similarity** between intermediate linear programs (LPs) appearing while solving the hierarchical multiobjective linear program (**h-MOLP**) for the decision of whether we should solve the current LP from **scratch** or use the **available feasible solution** obtained from the **previous** LP solve.

h-MOLP with bounded variables

$$\texttt{lexmin}\ {c^1}^{\mathsf{T}}x,\ {c^2}^{\mathsf{T}}x, \cdots,\ {c^t}^{\mathsf{T}}x$$
$$\text{subject to}\quad Ax = b,$$
$$l \le x \le u.$$

lexmin: ${c^1}^{\mathsf{T}}x$ is more important than ${c^2}^{\mathsf{T}}x$ which is, more important than ${c^3}^{\mathsf{T}}x$, and so on and, ${c^t}^{\mathsf{T}}x$ is of least importance.

Lexicographic Methods

1. LP^k - Constraint-addition rule
2. modLP^k - Variable fixing rule

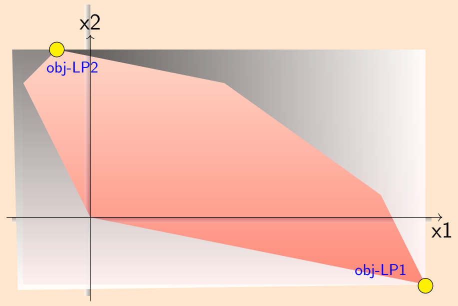
$$\begin{aligned}
\text{LP}^k := \min\ & c^k x & \qquad \text{modLP}^k := \min\ & c^k x \\
\text{s.t.}\quad & Ax = b, & \text{s.t.}\quad & Ax = b, \\
& c^i x = y^i,\ i = 1, \dots, k-1, & & x_j = f_j,\ j \in J^k \subseteq \{1, \dots, n\}, \\
& l \le x \le u. & & l \le x \le u.
\end{aligned}$$

Remarks

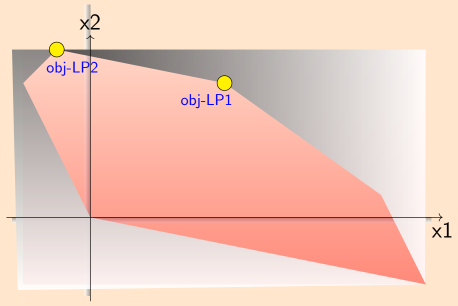
1. Both provide Pareto-optimal solution
2. Most literature describes LP^k rule in the discussion of the method
3. Most solvers prefer implementing the modLP^k
4. Assumption: h-MOLP is nontrivial - alternative optimal solutions while solving the underlying LPs, and we have conflicting objectives

Motivation

- To obtain **one solution** point both need the solution of **many LPs**
- To **speed it up**, many solvers provide a feature of **Reoptimization**
- However, a rule is required to **selectively decide** when to use Reoptimization



Prefer Solving from Scratch



Prefer Reoptimization

Notion of Similarity

LP^1 is **similar** to LP^2, if the solution of LP^1 helps solve LP^2 faster than just solving it from scratch.

Definition 1. *Let $LP^1 := \min\{c^1x \mid Ax = b,\ l^1 \le x \le u^1\}$ and $LP^2 := \min\{c^2x \mid Ax = b,\ l^2 \le x \le u^2\}$ are two LPs. We assume that the feasible sets of both LP^1 and LP^2 are non-empty. With respect to p and B, the optimal solution and the optimal basis of LP^1, we say LP^1 and LP^2 are "similar" if*

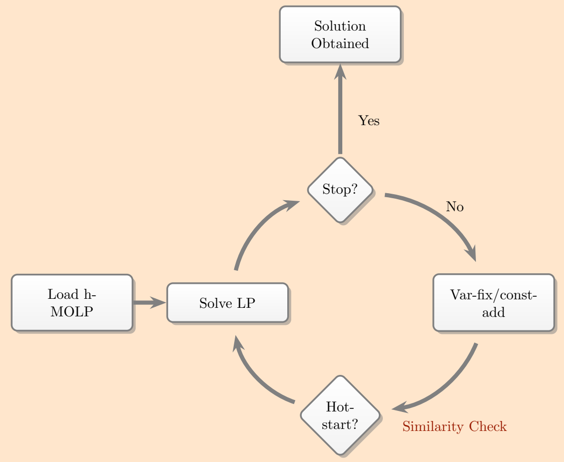
$$\left[1 - \left(\frac{\sum_{j=1}^{n} I_{p_j} |c_j^2 - c_j^1|}{\sum_{j=1}^{n} |c_j^2 - c_j^1|}\right)\right] \ge \kappa,$$

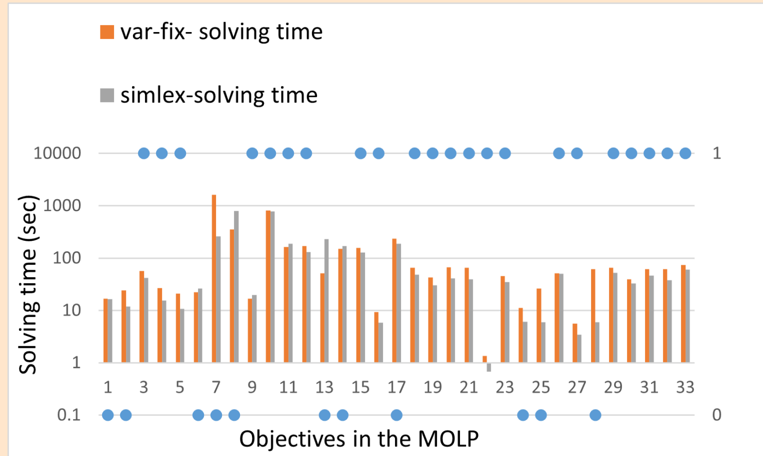
where $0 \le \kappa \le 1$ is the given parameter. I_{p_j} the indicator function is equal to 1 if $e_j \ne 0$ and $\overline{c_j^1} = 0$, else is set to 0. Here, reduced cost of LP^1 is $\overline{c^1} := c^1 - c_B^1 B^{-1} A$ and $e := c^2 - c_B^2 B^{-1} A$ is assumed as the estimated reduced cost of LP^2.

Result

Consider p, B and $\overline{c^1}$ are at optimality of LP^1. Consider a perturbed cost vector $c^2 = c^1 + \delta$ where $\delta \ne 0$ is given. Let $e = c^2 - c_B^2 B^{-1} A$. For any component $p_i, i \in \{1, \dots, n\}$, if $\overline{c_i^1} \cdot e_i \le 0$ and, $\overline{c_i^1}$ and e_i both can not together be zero, the basis B of LP^1 is not the optimal basis of LP^1 with perturbed cost vector c^2.

SimLex: Similarity-based Decision to Reoptimize





Performance comparison: SimLex Vs Variable-fixing. Blue points at level 1 indicate solving with hotstart and at level 0 indicate solving from scratch

Result and Future Work

1. Comparison is done over default CPLEX12.10 and other rules
2. Apply this idea with 1) **MOPLIB**, a library of benchmark multi objectives programs, and, 2) model of the **master production schedules** (MPS)
3. Reported **25%** (SGM-50) speedup from our rule over default CPLEX rule on obtaining MPS on **13** large scaled **consumer and products goods** (CPG) industry dataset
4. **Ideal parameter selection** of κ for similarity computation is not emphasized and is the main works that need to be done