Nantes Université

Computing Project

-

First year of Master's Degree in Optimization and Operations Research

LS2N - Laboratory of Digital Sciences of Nantes

Team MODELIS - Modélisation Optimisation et DEcision pour la Logistique, l'Industrie et les Services

---

MO-MILP Primal Heuristics

Improvement of Gravity Machine

Author:
Erwan Meunier[1]

Under the direction of:
Xavier Gandibleux[2]
Saïd Hanafi[3]

Defended on April 14, 2023

[1] erwan.meunier@etu.univ-nantes.fr
[2] xavier.gandibleux@univ-nantes.fr
[3] said.hanafi@uphf.fr

Contents

1 Abstract

2 Introduction

3 Definitions and Notations

We consider the problem as presented in [Gandibleux et al., 2021]. The multi-objective 0-1 linear optimisation problems with p objectives (p-01LP) considered can be formulated as follows:

$$\begin{aligned} \min z(x) &= Cx \\ \text{subject to } Ax &\leqq b \\ x &\in \{0,1\}^n \end{aligned}$$

where

- $x \in \{0,1\}^n$, the vector of n binary variables $x_j, j = 1, \ldots, n$;
- $A \in \mathbb{R}^{m \times n}$, the m constraints $A_i x \leq b_i,\ i = 1, \ldots, m$ and $b \in \mathbb{R}^m$;
- $C \in \mathbb{R}^{p \times n}$, the objective matrix where $p \geq 2$;
- $X := \{x \in \{0,1\}^n \mid Ax \leqq b\} \subseteq \mathbb{R}^n$, the set of feasible solutions, with $\mathbb{R}^n$ the decision space;
- $Y := \{Cx \mid x \in X\} \subseteq \mathbb{R}^p$, the outcome set, with $\mathbb{R}^p$ the objective space.

Thereafter, p is usually meant to be equal to 2 (*e.g.* we consider the bi-objective class of problems).

Definition 1 ([Ehrgott, 2005]). *A feasible solution $\hat{x} \in \mathcal{X}$ is called efficient or Pareto optimal if there is no other $x \in \mathcal{X}$ such that $f(x) \leq f(\hat{x})$. If $\hat{x}$ is efficient, $f(\hat{x})$ is called nondominated point. If $x^1, x^2 \in \mathcal{X}$ and $f(x^1) \leq f(x^2)$ we say x^1 dominates x^2 and $f(x^1)$ dominates $f(x^2)$. The set of all efficient solutions $\hat{x} \in \mathcal{X}$ is denoted $\mathcal{X}_E$ and called the efficient set. The set of all nondominated points $\hat{y} = f(\hat{x}) \in \mathcal{Y}$, where $\hat{x} \in \mathcal{X}_E$, is denoted $\mathcal{Y}_N$ and called the nondominated set.*

Definition 2 ([Ehrgott, 2005]). *A feasible solution $\hat{x} \in \mathcal{X}$ is called weakly efficient (weakly Pareto optimal) if there is no $x \in \mathcal{X}$ such that $f(x) < f(\hat{x})$, i.e. $f_k(x) < f_k(\hat{x})$ for all $k = 1, \ldots, p$. The point $\hat{y} = f(\hat{x})$ is then called weakly nondominated. A feasible solution $\hat{x} \in \mathcal{X}$ is called strictly efficient (strictly Pareto optimal) if there is no $x \in \mathcal{X}$,$x \neq \hat{x}$ such that $f(x) \leqq f(\hat{x})$. The weakly (strictly) efficient and nondominated sets are denoted $\mathcal{X}_{wE}(\mathcal{X}_{sE})$ and $\mathcal{Y}_{wE}$, respectively.*

Definition 3 (A lower bound set [Ehrgott and Gandibleux, 2001]). *A lower bound set for Y' is a subset $L \subseteq \mathbb{R}^p$*

1. *For each $y \in Y'$ there is some $l \in L$ such that $l \leqq y$*
2. *There is no pair $y \in Y', l \in L$ such that y dominates l.*

Definition 4 (An upper bound set [Ehrgott and Gandibleux, 2001]). *An upper bound set for Y' is a subset $U \subset \mathbb{R}^p$*

1. *For each $y \in Y'$ there is some $u \in U$ such that $y \leqq u$*
2. *There is no pair $y \in Y', \in U$ such that u dominates y.*

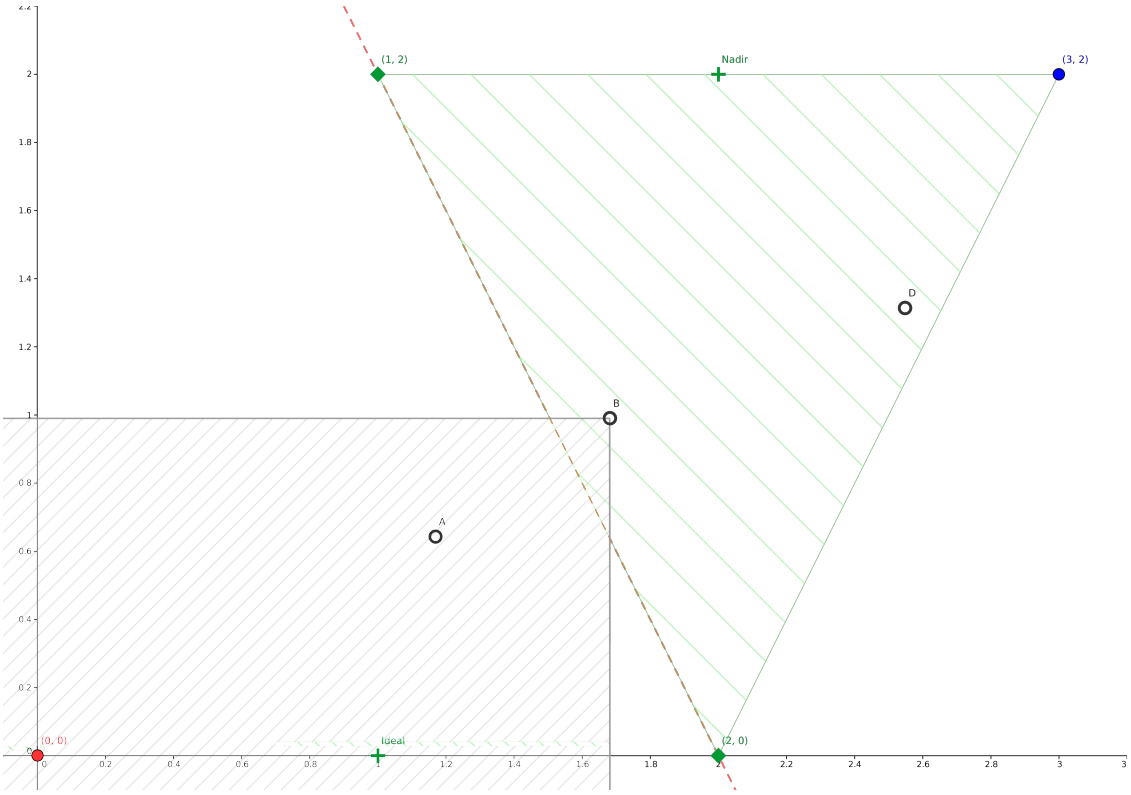

Figure 1: Projection of the solution space onto the cost space

Algorithm 1: Outline of Gravity Machine

Data: $\mathcal{D}$,n_L,maxTrial,maxTime
Result: L,U

begin
  $L, F \leftarrow$ ComputeGenerators$(\mathcal{D}, n_L)$;
  forall $z(\bar{x}^k) \in L, \bar{x}^k \notin F$ **do** //each k-th unfeasible gen
    timeStart $\leftarrow$ Time(); trial $\leftarrow 0$;
    timeout $\leftarrow$ false; feasible $\leftarrow$ false;
    $\tilde{x}, H, cycle \leftarrow$ RoundingSolution$(\bar{x}^k, H)$;
    while $\neg feasible \wedge \neg timeout$ **do**
      trial $\leftarrow$ trial $+1$;
      $\bar{x}, F$,feasible $\leftarrow$ ProjectSolution$(\tilde{x})$;
      if $\neg feasible$ **then**
        $\tilde{x}, H, cycle \leftarrow$ RoundingSolution$(\bar{x}, k, h)$;
        if *cycle* **then**
          $\tilde{x} \leftarrow$ PerturbSolution$(\tilde{x})$;
        end
      end
      timeout $\leftarrow$ (time()-timeStart $\geq$ maxTime) $\vee$ (trial=maxTrial);
    end
  end
  $U \leftarrow$ ExtractNonDominatedPoint(F);
end

4 Backgrounds

Gravity Machine is an algorithm aiming to compute an upperbound set for a multi-objective linear optimization problem with binary variables.

5 Soft constraint on cones

6 Improving generators

Generators are obtained by using the ϵ-constraints method which is one of the well-known scalarization techniques [Chankong and Haimes, 1983] aimed at getting a sample of efficient solutions of a MOLP. Those generators form an efficient set for the relaxed problem. That is to say, the set of all generators is also the lower bound set L. Since GM starts the search based on each k-th generator, one would like to have a better initial solution in order to reach a feasible solution faster.

6.1 Bug in Gravity Machine

The previous version of Gravity Machine did use the GLPK solver to generate the lower bound set by the ϵ-constraints method. However, closer to the line passing through the Nadir and the origin the more constrained the MILP representing the current generator. Since considered problems have lot of constraints and variables some numerical issues may arise [Gurobi Optimization, LLC, 2023]. Infeasibility hence may appear during the solving process for problems which has at least one (feasible) optimal solution. In addition, the *Julia* wrapper for GLPK seems to be known for returning the solution minimizing constraints violation[4]. Finally, as the termination status was not correctly handled, some generators turn out to be used whereas there were not "trully" feasible.

a primal feasibility tolerance attribute was hand-tuned to allow the solver to find a feasible solution. [Mittelman, 2023]

[EM] Stressing this point in the discussion

6.2 Empirical findings

Empirical evidences tend to show that very few variables are set to non-integral value for each generator.

6.3 Setting some variables binary

At first sight, non-solved to integrality variables should be set binary and the problem re-optimized. The latter allows GM to find initial solution Then, a generator closer to Y should result from a the re-optimization process. However, several issues arises:

6.3.1 Communicating vessels effect

As not all constraints are set binary, some variables found to be zero or one might become fractional. A number of variables found fractional after the second stage solving greater than the number of

[4] https://discourse.julialang.org/t/access-infeasible-optimization-results-glpk-with-semicontinuous-variable/59972

variables found fractional after the first stage solving represents the worst case. As far as our experimental data shows us, this situation is very rare.

6.3.2 Same generators lead to the same solution

Canonical cut constraint Provided a feasible solution s known to be binary, the following constraint allows the solving process to avoid returning s as a solution [Balas and Jeroslow, 1972]:

$$\sum_{j \in \{i | s_i = 0\}} x_j + \sum_{j \in \{i | s_i = 1\}} 1 - x_j \geq 1$$

The lower bound set is no longer guaranteed

Algorithm 2: Improving generators

Data: L,λ_1,λ_2,τ,α
Result: L_I,U

```
1  begin
2      L, F ← ComputeGenerators(D, n_L);
3      forall z(x̄^k) ∈ L, x̄^k ∉ F do //each k-th unfeasible gen
4          timeStart ← Time(); trial ← 0;
5          timeout ← false; feasible ← false;
6          x̃, H, cycle ← RoundingSolution(x̄^k, H);
7          while ¬feasible ∧ ¬timeout do
8              trial ← trial +1;
9              x̄, F,feasible ← ProjectSolution(x̃);
10             if ¬feasible then
11                 x̃, H, cycle ← RoundingSolution(x̄, k, h);
12                 if cycle then
13                     x̃ ← PerturbSolution(x̃);
14                 end
15             end
16             timeout ← (time()-timeStart ≥ maxTime) ∨ (trial=maxTrial);
17         end
18     end
19     U ← ExtractNonDominatedPoint(F);
20 end
```

Cones

6.3.3 Too much binary variables increase the solving time

As shown by our empirical data, less than 20

7 Improving projection

By the same token as the generators improvement, the projection used during GM can be improved by adding some integrity constraints on the non

8 Numerical experiments

Gravity Machine is implemented in Julia 1.8.5 [Bezanson et al., 2017]

8.1 Numerical instances

8.2 Quality measure

8.3 Numerical results

Experiments have been performed on-top of the following configuration:

- Laptop: Dell - XPS 13 9300
- OS: Linux Mint 21.1 Cinnamon - v5.6.8
- Kernel version: 5.19.0-38-generic
- Intel© Core™ i7-1065G7 CPU @ 1.30GHz × 4
- L1: 320 KiB, L2: 2 MiB, L3: 8 MiB
- RAM: 15.2 Go LDDR4
- SSD: 1024.2 Go

8.3.1

9 Conclusion and discussion

References

[Balas and Jeroslow, 1972] Balas, E. and Jeroslow, R. (1972). Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69.

[Bezanson et al., 2017] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.

[Chankong and Haimes, 1983] Chankong, V. and Haimes, Y. Y. (1983). Multiobjective decision making: Theory and methodology.

[Ehrgott, 2005] Ehrgott, M. (2005). *Multicriteria Optimization (2. ed.)*. Springer.

[Ehrgott and Gandibleux, 2001] Ehrgott, M. and Gandibleux, X. (2001). Bounds and bound sets for biobjective combinatorial optimization problems. In Köksalan, M. and Zionts, S., editors, *Multiple Criteria Decision Making in the New Millennium*, pages 241–253, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Gandibleux et al., 2021] Gandibleux, X., Gasnier, G., and Hanafi, S. (2021). A primal heuristic to compute an upper bound set for multi-objective 0-1 linear optimisation problems. Online. Springer.

[Gurobi Optimization, LLC, 2023] Gurobi Optimization, LLC (2023). Gurobi Guidelines for Numerical Issues - Gurobi Optimization gurobi optimizer reference manual. `https://www.gurobi.com/documentation/9.0/refman/num_grb_guidelines_for_num.html`. [Accessed 06-Apr-2023].

[Mittelman, 2023] Mittelman, H. (2023). Benchmarks for optimization software - milp benchmark. `http://plato.asu.edu/ftp/milp.html`. [Accessed 14-Apr-2023].