INFO-F424 - Combinatorial Optimization
Project - The p-Center Problem

Erica Berghman
Charles Hamesse

École Polytechnique de Bruxelles

May 2017

Abstract

The purpose of this project is to implement two formulations of the same combinatorial optimization problem in Julia, using the JuMP package. We will start by describing the mathematical aspects of both formulations, then we will explain our implementations and discuss their performance.

Contents

1 Introduction

1.1 Implementation

1.2 Compiling and running

Compiling

Running Some example runs:

```
code
```

2 Daskin (1995)

This is the formulation referred to as (P1) in the original paper.

2.1 Description

According to the usual canvas, the mathematical formulation is given as follows.

Variables Three variables are used in this formulation:

$$
\begin{aligned}
y_j &= \begin{cases} 1 & \text{if vertex } j \text{ is a center} \\ 0 & \text{otherwise} \end{cases} \\
x_{ij} &= \begin{cases} 1 & \text{if vertex } i \text{ assigns to a center in vertex } j \\ 0 & \text{otherwise} \end{cases} \\
p &= \text{maximum number of centers}
\end{aligned}
$$

Both indices i and j have a range of $[1, N]$ where N is the number of vertices of the instance.

Objective function

$$
\begin{aligned}
& min & z \quad (1) \\
\text{s.t.} \quad & \sum_{j \in N} d_{ij} x_{ij} \leq & z \quad (2)
\end{aligned}
$$

These two expressions ensure that the objective value is no less than the maximum vertex-to-center distance, which we want to minimize. Note that (2) is actually implemented as a constraint but shown here for the sake of readability.

Constraints

$$
\begin{aligned}
\sum_{j \in N} x_{ij} &= 1 \quad \forall i \in N & (3) \\
x_{ij} &\leq y_i \quad \forall i, j \in N & (4) \\
\sum_{j \in N} y_j &\leq p & (5) \\
y_j &\in \{0, 1\} \quad \forall j \in N & (6) \\
x_{ij} &\in \{0, 1\} \quad \forall i, j \in N & (7)
\end{aligned}
$$

Constraint (3) assigns each vertex to exactly one center. Constraint (4) ensures that no vertex assigns to v_j unless there is a center at v_j. Constraint (5) restricts the number of centers to p. Constraints (6) and (7) are the binary restrictions for variables x and y.

2.2 Implementation

2.3 Results

Example table:

Initial solution	Neighbourhood	Pivoting rule	ARCD	ACT
Random	Exchange	Best	36.50	23.43
Random	Exchange	First	42.66	9.70
Random	Insert	Best	3.57	24489.42
Random	Insert	First	4.20	7666.12
Random	Transpose	Best	37.57	108.45
Random	Transpose	First	28.59	72.03
Simplified-RZ	Exchange	Best	4.32	106.67
Simplified-RZ	Exchange	First	4.35	105.55
Simplified-RZ	Insert	Best	2.29	6782.88
Simplified-RZ	Insert	First	3.58	3176.73
Simplified-RZ	Transpose	Best	4.16	112.37
Simplified-RZ	Transpose	First	4.31	109.22

3 Calik and Tansel (2013)

This is the formulation referred to as (P3) in the original paper.

3.1 Description

3.2 Implementation

3.3 Results

4 Conclusion

4.1 Comparison of the two formulations

4.2 Wrap-up