A Dichotomic Algorithm for Transportation Network and Land Use Problem

Mamadou Koné[1], Mouhamadou A.M.T. Baldé[1], Babacar M. Ndiaye[1]

[1] Laboratory of Mathematics of Decision and Numerical Analysis, LMDAN-FASEG, University of Cheikh Anta Diop, BP 45087 Dakar-Fann, Dakar, Senegal

Correspondence: Babacar M. Ndiaye, Laboratory of Mathematics of Decision and Numerical Analysis, LMDAN-FASEG, University of Cheikh Anta Diop, BP 10700 Dakar-Fann, Dakar, Senegal

Received: November 11, 2018 Accepted: December 29, 2018 Online Published: January 22, 2018

doi:10.5539/jmr.v11n1p42 URL: https://doi.org/10.5539/jmr.v11n1p42

Abstract

Redeveloping sites to accommodate housing or new economic activities is a major urban policy challenge. This problem belongs to the class of NP-hard problems. In this paper, we present an attractive mixed integer nonlinear programming formulation for the transportation network and land use problem. We first introduce a new useful nonlinear formulation of this challenging combinatorial optimization problem. Then, an alternative to considering a linearity of the constraints is to reformulate the problem as a new exact, compact discrete linear model. The problem is solved by our new algorithm and numerical results are presented for a number of test problems in academics instances.

Keywords: land use, transportation network, assignment, algorithm, mixed integer programming

1. Introduction

The Transportation Network and Land-Use (TNLU) problem is a combination of the transportation network optimization problem and land use planning or Quadratic Assignment Problem (QAP). This model appears, for example, if we decided to find locations or relocations of some a city amenities (home, shop, work or leisure places), that may reduce travel time or travel distance. By solving this problem we want to have the best layout of the city in the direction of the location of activities and roads linking these activities. A better configuration of activity locations and transportation for a city would reduce congestion and pollution. For example, we can consider the Koopman and Beckmann (1957) version of the problem, which can informally be stated with reference to the following practical situation: a municipality must decide to assign n activities to an equal number of locations and want to minimize the total cost of location and transportation. For each pair of activities (i, j) a flow of communication f_{ij} is known, and for each pair of locations (l, k) the corresponding distance L_{kl} is known. The transportation cost between activities i and j is $f_{ij}L_{kl}$, where i and j are assigned to locations k and l, respectively. The objective of the municipality is to find (i) an assignment that minimizes the sum of the total linear cost for installing an activity to a location, (ii) the total quadratic interaction cost, (iii) the flow between the activities and (iv) total cost that link roads between activities.

Since the 60s many authors are interested in these two problems separately: for example Scott (1969), Boyce, Farhi and Weischedel (1973)and Hoang Hai Hoc (1973) for network design problems; Lawler (1963), Maniezzo (1997), Xia and Yuan (2006), Huizhen Zhang (2010) and many others for the activities's location problems. Marc Los and more recently Lin and Feng are interested in the combined problem. They have considered a more complex model in the sense that their formulation allows the location of several activities in the same area. Marc Los gave an exact solving method and heuristic methods. The Los's model take into account to the OD flow, once the activities have been fixed, via the shortest paths in the sense of a fixed cost independent of the flow on the arcs (see Gueye (2012)). For more details or a review of the literature, we refer the reader to Baldé and Ndiaye (2016), Duranton and Puga (2015), and Gueye (2012). The TNLU can be formulated as a Mixed Integer NonLinear Programming (MINLP) problem. Many algorithms and softwares (as Cplex (2016), Gurobi (2016), Couenne (2018), etc.) are proposed in the literature to solve non linear (quadratic) programmming problem, but present some limitations for large number of variables. The quadratic assignment problem constitutes the natural formulation of a large number of concrete problems, belonging to various fields. Its complexity is such that the proposed methods so far to solve it were not of sufficient effectiveness.

In this paper, we propose finite linearization and an algorithm to find good solutions with fewer iterations and running times. It is a new resolution method based on a dichotomic translation of a hyperplane and can give better results compared to those obtained by other methods of solvers. We refer the reader to see first Lavallée, Ndiaye and Seck (2011); and Ndiaye, Lavallée and Seck (2013) for an application of the algorithm to the Klee and Minty problem, for example. As the algorithm is designed to solve a linear programming problem, we first linearize the TNLU problem and introduce

(arbitrary constraint) support hyperplane guaranteed by the objective function. In other words, we add in the constraint a hyperplane parameterized by real α_0, which serves as a barrier. Secondly, to start with the resolution, we set an initial condition and a stopping criterion. To verify the obtained solution of the senegaulois algorithm, we compare it with solutions given by Cplex and Gurobi (we give to Cplex and Gurobi the quadratic problem).

The paper is organized as follows. In section 2 we present definition and assumptions. In section 3, we give the formulation of the TNLU. In section 4 we introduce a new linearization for the TNLU. In section 5 we develop a brief presentation of senegaulois algorithm. The section 6 provides numerical simulations using test problems in academics instances. Finally, in section 7, we provide conclusions and suggestions for future works.

2. Definitions, Assumptions and Theorems

Let us formulate the main concepts used in the paper, mainly for sections 3 and 5.

Definition 2.1 (Construction graph)**.** Let $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$ two non-oriented graphs, where N_1 and N_2 ($|N_1| = |N_2| = n$) represent respectively the set of vertices of the graph G_1 and G_2, and E_1 and E_2 represent respectively all edges (or arcs) of G_1 and G_2.

The graph matching plays a central role in solving correspondence problems in computer vision. Graph matching problems that incorporate pair-wise constraints can be cast as a quadratic assignment problem (QAP) (see Koopmans and Beckmann(1957)). Unfortunately, (QAP) is known to be NP-hard (see Sahni and Gonzalez (1976)) and many algorithms have been proposed to solve different relaxations.

Definition 2.2 (Exact Graph Matching)**.** The graph matching problem can be stated as follows: Given two graphs $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$, with $|N_1| = |N_2|$ the problem is to find a one-to-one mapping $\pi : N_1 \longrightarrow N_2$ such that $(i, j) \in E_1$ if and only if $(\pi(i), \pi(j)) \in E_2$. When such a mapping π exists, this is called an isomorphism.

Definition 2.3 (Isomorphic graphs)**.** Two graphs G_1, G_2 are isomorphic (symbol $\cong$) if there is a bijection $\pi : N(G_1) \to N(G_2) : (i, j) \in E(G_1) \Longleftrightarrow (\pi(i), \pi(j)) \in E(G_2)$.

A bijection of a finite set of cardinal n on itself is a *permutations* of n elements. We denote by $S_n$the set of *permutations* de $\{1, ..., n\}$.

$$S_n = \{\pi : N_1 \longrightarrow N_2 \mid \pi \text{ is bijective}\}$$

Given an activity $i \in E(G_1)$, then $\pi(i) = k$ implies that activity i ($i-$th component in the ordered set N_2) is at location k. In the optimization problem, we are looking for the best assignment, i.e., we have to optimize some objective function which depends on the assignment π. Assignments can be represented in different ways. The bijective mapping between two finite sets N_1 and N_2 can be represented in a straight forward way by a perfect matching in a bipartite graph $G = (N_1, N_2; E)$, where the vertex sets N_1 and N_2 have n vertices. Edge $(i, k) \in E$ is an edge of perfect matching iff $k = \pi(i)$.

Definition 2.4 (Final graph)**.** Let $G = (N, E)$ be the general graph. G is determined by the data of two sets:

- a non-empty finite set N whose elements are called vertices

$$N = \Big\{(i, \pi(i)) : i \in N(G_1), \pi(i) \in N(G_2)\Big\}$$

- a set E of vertex pairs called edges.

$$E = \Big\{((i, k), (j, l)) : (i, j) \in E(G_1), (\pi(i), \pi(j)) \in E(G_2), i \neq j, k \neq l\Big\}$$

  The arcs $((i, k), (j, l))$ for all $i \neq j, k \neq l$ of G are connected if starting from the nodes i to the zone $k = \pi(i)$ it is possible to reach the node j at the zone $l = \pi(j)$. We can see, in the Figure 1 the location of the set of activities on the zones as a bijective **mapping** $\pi(.)$ of all activities to all zones.

Now, let's define two theorems dealing with the question of existence of a hyperplane that separates two given disjoint convex subsets.

Definition 2.5. We define hyperplane $\mathcal{H}$ with equation $[f = \alpha]$ the linear set :

$$\mathcal{H} = \{x \in \mathbb{R}^n : f(x) = \alpha\} \tag{1}$$

Theorem 1 (Hahn-Banach I)**.** Let $\mathbb{E}$ be a finite-dimensional vector space with inner product $\langle .,. \rangle$. Let $C_1, C_2 \subset \mathbb{E}$ two non-empty convex sets such that $C_1^\infty \cap C_2^\infty = \{0\}$ (where C_1^∞ and C_2^∞ are asymptotic cones of C_1 and C_2, respectively). Then, it is possible to separate C_1 and C_2. There exists a vector $\xi \in \mathbb{E}$ such that:

$$\sup_{x_1 \in C_1} \langle \xi, x_1 \rangle \leq \inf_{x_2 \in C_2} \langle \xi, x_2 \rangle$$

Corollary 1. Suppose that $C_1, C_2 \subset \mathbb{E}$ are non-empty closed convex sets with C_1 compact, C_2 closed. If $C_1 \cap C_2 = \emptyset$ then C_1 and C_2 are strictly separated by a hyperplane.

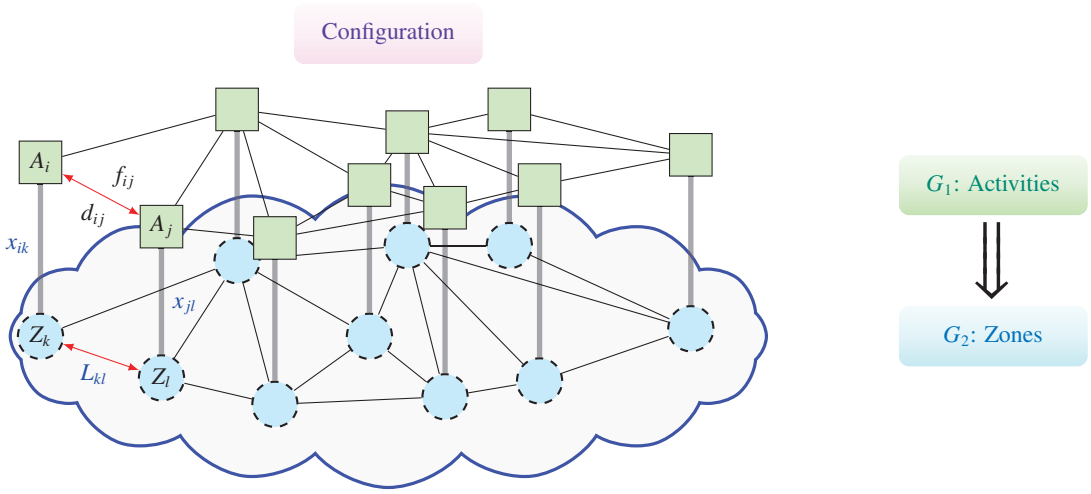

Figure 1. Localization area and network design

Theorem 2 (Hahn-Banach II)**.** Let $\mathbb{E}$ be a finite-dimensional vector space with inner product $\langle .,. \rangle$. Let $C_1, C_2 \subset \mathbb{E}$ be two non-empty disjoint convex subsets. Then, there exists a nonzero vector $\xi \in \mathbb{E} \backslash \{0\}$ such that :

$$\sup_{x_1 \in C_1} \langle \xi, x_1 \rangle \leq \inf_{x_2 \in C_2} \langle \xi, x_2 \rangle$$

Corollary 2.

- Let C be a non-empty closed convex set. $\forall u \in \partial C$ (the border of C), C is supported at u. If C is a convex set of a topological vector space, any point of the border of C is in a supporting hyperplane.
- Let C be a closed convex set, then
$$C = \bigcap_{\Pi \in p} \Pi$$
where p is the set of the half-space containing C.

More details can be found in Lavallée, Ndiaye and Seck (2011); and Ndiaye, Lavallée and Seck (2013).

3. Formulation of the Mathematical Model

3.1 Indices, Sets and Parameters

- N_1: set of activities (= $\{1, ..., n\}$),
- N_2: set of zones (= $\{1, ..., n\}$),
- N: set of assignment,
- $E_1 = \{(i, j) : i, j \in N_1, \ \ i \neq j\}$: set of arcs between each activity pairs,

- $E_2 = \{(k,l) : k,l \in N_2,\ \ k \neq l\}$: set of arcs between each pairs of the area,
- $E = \{((i,k),(j,l)) : (i,j) \in E_1, (k,l) \in E_2,\ i \neq j,\ k \neq l\}$: set of edges,
- $F = (f_{ij})_{n\times n}$: the flow matrix volume from activity i and j,
- $D = (d_{ij})_{n\times n}$: the distance matrix between location i and j,
- $C = (c_{ik})_{n\times n}$: location cost matrix of the activity i on the area k,
- $B = (b_{ij})_{n\times n}$: construction cost matrix of "build" the link or road (i, j).

3.2 Variables

The variables of the model can be divided into three groups. The first group includes the binary variables x_{ik}. The second group y_{ij} represents the network configuration. The last group contains the continuous variables L_{kl} that determine the length of the path k to l.

- Let us start with the observation that every permutation π of the set $N = \{1, ..., n\}$ can be represented by an $n \times n$ matrix $X_\pi = (x_{ij})$, such that

$$x_{ik} = \begin{cases} 1 & \text{if activity } i \text{ is assigned to area } k\ (\pi(i) = k), \\ 0 & \text{otherwise } (\pi(i) \neq k). \end{cases} \tag{2}$$

  Matrix $X_\pi = (x_{ij})$ is called a *permutation matrix* and caracterized by following *assignment contraints*:

$$\sum_{k:(i,k)\in E} x_{ik} = 1, \quad \text{for all } i \in N_1 \tag{3}$$

$$\sum_{i:(i,k)\in E} x_{ik} = 1, \quad \text{for all } k \in N_2 \tag{4}$$

$$x_{ik} \in \{0,1\}, \quad \text{for all } (i,k) \in E.$$

  The 3 and 4 equalities of assigning each element of N_1 to an element of N_2.

- To determine the existence of a path in the network (whether or not there is a direct road link between i and j), we define binary variables (boolean) y_{ij} here below:

$$y_{ij} = \begin{cases} 1 & \text{if the link } (i,j) \text{ is selected for construction,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

- Then it is necessary to introduce variables of shortest path L_{kl} between the location k and location l and variables z_{ij}^{kl} which take value 1 if the shortest path between k and l goes by link (i, j) and 0 otherwise. We have:

$$z_{ij}^{kl} = \begin{cases} 1 & \text{if the link } (i,j) \text{ is selected in the path from } k \text{ to } l \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

If we consider the distances of the arcs (i.e. between each pair of activity) as the costs associated with the flow on the arcs, then the problem of the *shortest path* from k to l can be formulated as a minimum-cost flow problem (FCM) to move a flow unit from the activity i to k to the activity j to l.

The *out-degree* of i: $\delta^+(i) = \left\{j \in N_1 : (i,j) \in E_1\right\}$ is the number of arcs that have i as an initial vertex (e.g. the number of arcs coming out of i).

The *in-degree* of i: $\delta^-(i) = \left\{j \in N_1 : (j,i) \in E_1\right\}$ is the number of arcs that have i as a terminal vertex (e.g. the number of arcs coming into i).

The minimum-cost network flow problem (FCM) is a fundamental problem in network analysis that involves sending flow over a network at minimal cost. Let $G_1 = (N_1, E_1)$ be a directed graph. For each link $(i, j) \in E_1$, associate a cost per unit of flow, designated by d_{ij}.

$$(FCM): \quad \min_{\pi \in S_n, z} \sum_{(i,j)\in E_1} \sum_{(k,l)\in E_2} d_{ij} z_{ij}^{kl} \tag{7}$$

$$s.t.: \sum_{j:(i,j)\in\delta^+(i)} z_{ij}^{kl} - \sum_{j:(j,i)\in\delta^-(i)} z_{ji}^{kl} = \begin{cases} 1 & \text{si} \quad i = o(k) \\ 0 & \text{si} \quad i \in N \setminus \{o(k), d(l)\} \\ -1 & \text{si} \quad i = d(l) \end{cases} \tag{8}$$

$$z_{ij}^{kl} \in \{0, 1\}, \qquad (i, j) \in E_1, (k, l) \in E_2. \tag{9}$$

where, $\mathbf{o}(k)$: is **origin** (area of origin) and $\mathbf{d}(l)$: is **destination** (area of destination) .

The shortest-path problem (SPP) (Kelly and O'Neill (1991))is to find the directed paths of shortest length from a given root node to all other nodes. Moreover, the (SPP) is also a special case of the (FCM), where the objective is to find the minimum distance or length between two given nodes (e.g., nodes k and l). Here, the values of the $n \times 1$ vector b are now restricted only to 0, 1 or -1. The mathematical formulation of the (SPP) is as follows:

$$(SPP): \quad \min_{(k,l)\in E_2} L_{kl} \tag{10}$$

$$s.t.: \sum_{(i,j)\in E_1} d_{ij} z_{ij}^{kl} \leq \ L_{kl}, \quad (k, l) \in E_2 \tag{11}$$

$$\sum_{j:(i,j)\in\delta^+(i)} z_{ij}^{kl} - \sum_{j:(j,i)\in\delta^-(i)} z_{ji}^{kl} = b(i) = \begin{cases} 1 & \text{si} \quad i = o(k) \\ 0 & \text{si} \quad i \in N \setminus \{o(k), d(l)\} \\ -1 & \text{si} \quad i = d(l) \end{cases} \tag{12}$$

$$z_{ij}^{kl} \in \{0, 1\}, \qquad (i, j) \in E_1, (k, l) \in E_2. \tag{13}$$

The optimum solution to this problem sends unit flow from the root to every other node along a shortest path.

3.3 Transportation Network and Land Use (TNLU) Problem

This problem is a combined model composed of the following subproblems: an activity localization problem for the optimal location of activities; a problem of configuration and dimensioning of the transport network to define a transport network or its evolutions, considering a set of potential road, having a cost of construction and use, according to the observed flows. The objective is to assign each activity to a location such that the total cost is minimized:

The constraints 3.14 and 3.15 are the so-called assignment polytope or assignment constraints (see Burkard, Cela, Pardalos and Pitsoulis (1999)), they make it possible to ensure that each activity is assigned in a zone and each zone accommodates one and only one activity. Equalities 3.16 and 3.17 define the origin and destination of the path. The constraint 3.18 is the conservation constraint at each node. The constraints 3.19 ensure that the sum of the distances of the activities assigned to the different zones does not exceed the maximum distance (or length) L_{kl} between the zones. The constraints 3.20 make it possible to define the roads to construires or not. The 3.21 and 3.22 constraints ensure that the decision variable is binary. The 3.23 constraints ensure that the decision variables are continuous.

4. Linearizations of the QAP

To linearize the objective function we introduce new variables and linear contraints for the QAP to eliminate the quadratic term of 3.13. This methods transform the problem into a mixed linear program (see Burkard, Cela, Pardalos and Pitsoulis (1999)). The first linearization, proposed independently by several authors Fortet (1960), Watters (1967) , involved the addition of one binary variable and two linear constraints for each product of two variables. In 1974, Glover and Woolsey (1974) made a major breakthrough by proposing a linearization where the additional variable was not required to be explicitly defined as an integer. This linearization will be referred to, in this paper, as the *standard linearization*. Consider the problem of assigning a set of activities to a set of locations, with the cost being a function of the distance and flow between the activities, plus costs associated with a facility being placed at a certain location. Hence the name *Quadratic Assignment Problem* by Koopmans and Beckmann (1957):

$$(\mathbf{QAP}): \quad \underset{x,L}{\text{Minimize}} \sum_{(i,k)\in E} c_{ik} x_{ik} + \sum_{((i,k),(j,l))\in E} f_{ij} L_{kl} x_{ik} x_{jl} \tag{14}$$

$$s.t. \ \ x \in X, \ \ x \ \text{ binary}. \tag{15}$$

MINLP: **Mixed Integer NonLinear Problem**

$$\underset{x,y,L}{\text{minimize}} \sum_{(i,k)\in E} c_{ik}x_{ik} + \sum_{((i,k),(j,l))\in E} f_{ij}L_{kl}x_{ik}x_{jl} + \sum_{(i,j)\in E_1} b_{ij}y_{ij} \tag{3.13}$$

$s.t.:$

$$\sum_{i:(i,k)\in E} x_{ik} = 1, \qquad \forall\, k \in N_2, \tag{3.14}$$

$$\sum_{k:(i,k)\in E} x_{ik} = 1, \qquad \forall\, i \in N_1, \tag{3.15}$$

$$\sum_{j:(i,j)\in E_1} z_{kj}^{kl} = 1, \qquad \forall\, (k,l) \in E_2, \tag{3.16}$$

$$\sum_{i:(j,i)\in E_1} z_{il}^{kl} = 1, \qquad \forall\, (k,l) \in E_2, \tag{3.17}$$

$$\sum_{j:(i,j)\in E_1} z_{ij}^{kl} = \sum_{j:(j,i)\in E_1} z_{ji}^{kl}, \qquad \forall\, (k,l) \in E_2, \forall\, i \in N_1 \tag{3.18}$$

$$\sum_{(i,j)\in E_1} d_{ij}z_{ij}^{kl} \le L_{kl}, \qquad \forall\, (k,l) \in E_2, \tag{3.19}$$

$$z_{ij}^{kl} + z_{ji}^{kl} \le y_{ij}, \qquad \forall\, (i,j) \in E_1, (k,l) \in E_2, \tag{3.20}$$

$$x_{ik}, y_{ij} \in \{0,1\}, \qquad \forall\, (i,k) \in E, (i,j) \in E_1, \tag{3.21}$$

$$z_{ij}^{kl} \in \{0,1\}, \qquad \forall\, (i,j) \in E_1, (k,l) \in E_2, \tag{3.22}$$

$$L_{kl} \ge 0, \qquad \forall\, (k,l) \in E_2. \tag{3.23}$$

Here the feasible region X is the *assignment polytope*.

$$X = \left\{ x_{ik} \in \{0,1\} : \sum_{i:(i,k)\in E} x_{ik} = 1 \quad \text{for all} \quad k \in N_2, \quad \sum_{k:(i,k)\in E} x_{ik} = 1 \quad \text{for all} \quad i \in N_1 \right\} \tag{16}$$

4.1 Linearization of the Product of Two Variables

4.1.1 Standard Linearization ($Bin \times Bin$)

Standard Linearization (SL) consists of linearizing the product of two binary variables (see Glover (1975)). The principle of "standard" (or "classic") linearization is to replace remplace $x_{ik}x_{jl}$ par X_{ikjl}. Then add the three inequalities below:

$$(SL): \begin{cases} X_{ikjl} \le x_{ik}, \\ X_{ikjl} \le x_{jl}, \\ X_{ikjl} \ge x_{ik} + x_{jl} - 1, \\ X_{ikjl} \ge 0 \qquad \forall\, ((i,k),(j,l)) \in E, \\ x_{ik} \in \{0,1\} \qquad \forall\, (i,k) \in E. \end{cases} \tag{17}$$

The problem (**QAP**) is then rewritten thus adding to the assignment constraint the system 17:

$$(\mathbf{QAP}): \begin{cases} \underset{L,X}{\text{Minimize}} \sum_{(i,k)\in E} c_{ik}x_{ik} + \sum_{((i,k),(j,l))\in E} f_{ij}L_{kl}X_{ikjl} \\ \qquad\qquad \text{subject to} \quad 15, -16. \end{cases} \tag{18}$$

4.1.2 Product Linearization ($Bin \times Bin$)

We always consider the formulation of Koopmans and Beckmann (1957). We multiply both sets of assignment constraints 4.3 by each problem variable x_{ik}, $\forall\ (i,k) \in E$. We then linearize the resulting products by requiring $X_{ikjl} = x_{ik}x_{jl}$, $\forall\ ((i,k),(j,l)) \in E$. We obtain the following linearization constraints (see Blanchard, Elloumi, Faye and Wicker (2002):

$$\begin{cases} \sum\limits_i x_{ik}x_{jl} = x_{jl} \\ \sum\limits_k x_{ik}x_{jl} = x_{jl} \end{cases} \Longrightarrow \qquad (X_{ikjl} = x_{ik}x_{jl}) \qquad \Longrightarrow \begin{cases} \sum\limits_i X_{ikjl} = x_{jl} \\ \sum\limits_k X_{ikjl} = x_{jl} \end{cases} \tag{19}$$

The reformulation of the problem (**QAP**) is given as follows:

$$\begin{cases} \underset{X,L}{\text{Minimize}} & \sum\limits_{(i,k)\in E} c_{ik}x_{ik} + \sum\limits_{((i,k),(j,l))\in E} f_{ij}L_{kl}X_{ikjl} \\ s.t.: & \sum\limits_{i:(i,k)\in E} X_{ikjl} = x_{jl}, \quad \forall\ j \in N_1,\ \forall\ k,l \in N_2 \\ & \sum\limits_{k:(i,k)\in E} X_{ikjl} = x_{jl}, \quad \forall\ i,j \in N_1,\ \forall\ l \in N_2 \\ & X_{ikjl} \geq 0 \qquad \forall\ ((i,k),(j,l)) \in E \\ & x_{ik} \in \{0,1\} \qquad \forall\ (i,k) \in E. \end{cases} \tag{20}$$

4.1.3 Linearizing the Product of a Binary and a Continuous Variable ($Bin \times Cont$)

In the objective-function, the product $L_{kl}X_{ikjl}$ is replaced by a non-negative variable Z_{ikjl} which will take the same value as L_{kl} when $X_{ikjl} = 1$, and will be zero when $X_{ikjl} = 0$. If L_{kl} is bounded below by zero and above by $L_{\max}$ (or any Big$M$), i.e. $0 \leq L_{kl} \leq L_{\max}$. Suppose now that we have $Z_{ikjl} = L_{kl}X_{ikjl}$, where L_{kl} is a continuous variable and X_{ikjl} is your binary variable. Add the three inequalities below (Coelho (2018))

$$Z_{ikjl} = L_{kl}X_{ikjl} \Longrightarrow \begin{cases} Z_{ikjl} \leq L_{\max}X_{ikjl}, \\ Z_{ikjl} \leq L_{kl}, \\ Z_{ikjl} \geq L_{kl} - (1 - X_{ikjl})L_{\max}, \\ Z_{ikjl} \geq 0, \end{cases} \tag{21}$$

Note that if X_{ikjl} is zero, than the first inequality ensures that Z_{ikjl} will be zero as well (note that the third inequality only states that Z_{ikjl} has to be greater than a negative number) . On the other hand, if X_{ikjl} is 1, the first inequality ensures that Z_{ikjl} is less than our Big$M = L_{\max}$, which is further tightened by our second inequality. The last inequality states then that Z_{ikjl} has to be greater than or equal to $L_{\max}$, exactly as we wanted. Finally, if $L_{\max}$ is not bounded below by 0, but has bounds $[0, L_{\max}]$ (with $L_{\max}$ assumed to be positive due to the last inequality), then the form of the linearized inequalities are the following:

4.2 Binary to Continuous Variables

Consider a problem $(IP)_{0-1}$ involving a binary variable $x \in \{0,1\}$. This can be reformulated as follows (see Billionnet, Elloumi and Lambert (2013)):

$$(IP)_{0-1}: \begin{cases} \underset{x}{\text{Min}}\ f(x) \\ \text{s.t.} \\ x_i \in X \subseteq \{0,1\},\ i = 1,...,n. \end{cases} \qquad \Longleftrightarrow \qquad \begin{cases} \min\limits_x f(x) \\ \text{s.t.} \\ x_i^2 - x_i = 0,\ i = 1,...,n. \end{cases} \tag{22}$$

Since a binary variable $x_i \in X$ can only take values in $\{0,1\}$, any univariate equation in x that has exactly $x = 0$ and $x = 1$ as solutions can replace the binary constraint $x_i \in \{0,1\}^n$. The most commonly used is the quadratic constraint $x_i^2 = x_i$. The linear reformulation of $(IP)_{0-1}$ consists of asking $X_{ii} = x_ix_i$, $\forall\ i$, and then using standard linearization 4.1.1.

Additionally, for all $x_i \in \{0,1\},\ i = 1,...,n, X_{ii} = x_i,\ \forall\, i.$

$$\begin{cases} X_{ii} \le x_i, \\ X_{ii} \ge x_i, \\ X_{ii} \ge x_i + x_i - 1, \\ X_{ii} \ge 0\ \ \forall\, i, j, \\ x_i \in \{0,1\}\ \ \forall\, i. \end{cases} \iff \begin{cases} X_{ii} \le x_i, & (a) \\ X_{ii} \ge x_i, & (b) \\ X_{ii} \ge 2x_i - 1, & (c) \\ X_{ii} \ge 0\ \ \forall\, i & (d) \end{cases} \tag{23}$$

Constraints (a) and (b) imply $X_{ii} = x_i$. Consequently, contraints (c) and (d) become $0 \le \mathtt{X_{ii}} \le 1$.

This process would reduce all binary problems to nonconvex quadratically constrained problems. We refer the reader to Liberti, Cafieri and Tarissan (2009) for more details.

Finally, we obtain the continue linearization problem:

5. The Senegaulois Algorithm

We consider a linear programming problem below :

$$\begin{aligned} &\max\ c^T x \\ &s.c.\quad Ax \le b \\ &\qquad\ x \ge 0 \end{aligned} \tag{24}$$

where x represents the vector of variables (to be determined), c and b are vectors of (known) coefficients, A is a (known) matrix of coefficients, and $(.)^T$ is the matrix transpose. Recall that convex polyhedron on $\mathbb{E}$ is a set P as:

$$P = \{x \in \mathbb{R}^n : Ax \le b\}, \tag{25}$$

Let f be a linear functional on $\mathbb{E}$, $\alpha \in \mathbb{R}$, $C_1 \subset \mathbb{E}$ and $C_2 \subset \mathbb{E}$.

$$f(x) = c^T x,\ x \in \mathbb{R}^n$$

Now we briefly recall some concepts, such as the **Hahn-Banach theorems**. The proofs can be obtained in Achmanov (1984), and in Brezis, Ciarlet and Lions (1999) for polyhedron.

Using the separation theorem, every such hyperplane $\mathcal{H}$ divides the whole space in two convex sets:

$$\mathcal{H}_l = \{x \in \mathbb{R}^n : f(x) \le \alpha\}$$

and

$$\mathcal{H}_r = \{x \in \mathbb{R}^n : f(x) \ge \alpha\}$$

known as *halfspaces*. One part contains the polytope K defined from the constraints in 1.

Then, the following cases hold:

1. Hyperplane $\mathcal{H}$ does not divide the whole polytope K of constraints, then $\mathcal{H}$ is outside the feasible solutions, *i.e.* $\mathcal{H} \cap K = \emptyset$ ($\mathcal{H} \cap K$ does not contain any solution). In such case, it is necessary to choose another hyperplane $\mathcal{H}' \,//\, \mathcal{H}$ (*parallel*) such that $\mathcal{H}'$ lies between the origin 0 and $\mathcal{H}$.
2. Hyperplane $\mathcal{H}$ divides the whole polytope K of constraints, then there is a non-empty intersection between $\mathcal{H}$ and K, *i.e.* $\mathcal{H} \cap K \ne \emptyset$. In this situation, the problem contains some feasible solutions and K is separated by $\mathcal{H}$. If the stopping test fails, then a new separation hyperplane $\mathcal{H}^* \,//\, \mathcal{H}$ is chosen such that $\mathcal{H}^*$ lies in the half-space which does not contain the origin 0.
3. Polytope K is empty, the problem has no solution.

5.1 The Algorithm

Suppose $\mathcal{H} : f(x) = \alpha$ and K the initial polytope defined by the set of constraints of problem 5.1. If $H \cap K = \emptyset$, then we state $\alpha := \dfrac{\alpha}{2}$. That is to say that we divide the gap between 0 and α into two parts. It is the main idea of the algorithm.

Now suppose that we have two values of α denoted by α_0 and α_1 such that:

MILP1: **Mixed Integer Linear Problem with continuous variables**

$$\underset{x,Z,y}{\text{minimize}} \sum_{(i,k)\in E} c_{ik}x_{ik} + \sum_{((i,k),(j,l))\in E} f_{ij}Z_{ikjl} + \sum_{(i,j)\in E_1} b_{ij}y_{ij}$$

Subject to:

Assignment

$$\sum_i x_{ik} = 1, \qquad \forall k \in N_2, \tag{4.11}$$

$$\sum_k x_{ik} = 1, \qquad \forall i \in N_1, \tag{4.12}$$

Conservation of flows for each pair (k, l)

$$\sum_{j:(i,j)\in E_1} z_{kj}^{kl} = 1, \qquad (k,l) \in E_2, \tag{4.13}$$

$$\sum_{i:(j,i)\in E_1} z_{il}^{kl} = 1, \qquad (k,l) \in E_2, \tag{4.14}$$

$$\sum_{j:(i,j)\in E_1} z_{ij}^{kl} = \sum_{j:(j,i)\in E_1} z_{ji}^{kl}, \qquad (k,l) \in E_2, \tag{4.15}$$

$$\sum_{(i,j)\in E_1} d_{ij}z_{ij}^{kl} \le L_{kl}, \qquad \forall\,(k,l) \in E_2, \tag{4.16}$$

Links

$$z_{ij}^{kl} + z_{ji}^{kl} \le y_{ij}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.17}$$

Linearization $(Bin \times Bin) \implies X_{ikjl} = x_{ik}x_{jl}$

$$X_{ikjl} \le x_{ik}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.18}$$

$$X_{ikjl} \le x_{jl}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.19}$$

$$X_{ikjl} \ge x_{ik} + x_{jl} - 1, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.20}$$

Linearization $(Bin \times Cont) \implies Z_{ikjl} = L_{kl}X_{ikjl}$

$$Z_{ikjl} \le L_{\max}X_{ikjl}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.21}$$

$$Z_{ikjl} \le L_{kl}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.22}$$

$$Z_{ikjl} \ge L_{kl} - (1 - X_{ikjl})L_{\max}, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.23}$$

The variables

$$0 \le x_{ik}, y_{ij} \le 1, \qquad \forall\,(i,k) \in E, (i,j) \in E_1, \tag{4.24}$$

$$0 \le z_{ij}^{kl} \le 1, \qquad \forall\,(i,j) \in E_1, (k,l) \in E_2, \tag{4.25}$$

$$L_{kl} \ge 0, \qquad \forall\,(k,l) \in E_2. \tag{4.26}$$

$$Z_{ikjl} \ge 0. \qquad \forall\,(i,k) \in E; (i,j) \in E_1, \tag{4.27}$$

- $\mathcal{H} : f(x) = \alpha_0$ and $K \cap \mathcal{H} \neq \emptyset$
- $\mathcal{H} : f(x) = \alpha_1$ and $K \cap \mathcal{H} = \emptyset$

Then we take a new value of

$$\alpha = \frac{\alpha_0 + \alpha_1}{2} \tag{26}$$

In addition, two cases are possible :

1. $\mathcal{H} : f(x) = \alpha$ and $K \cap \mathcal{H} = \emptyset$, then put $\alpha_1 := \alpha$
2. $\mathcal{H} : f(x) = \alpha$ and $K \cap \mathcal{H} \neq \emptyset$, then put $\alpha_0 := \alpha$

And so on.
In this algorithm the hyperplane is defined by:

$$\mathcal{H} = \{x \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} c_i x_i = \alpha\} \tag{27}$$

where $\sum_{i=1}^{n} c_i x_i$ is the objective function (the one to maximize in our purpose) of the LP. **Remark:** From a geometrical point of view, at each step of the algorithm, the new hyperplane is obtained from the previous one by a translation. Thus, we search vectors that characterizes polytope $\mathcal{H}$ defined by (27); then we search an unitary vector $\overrightarrow{v}$ as vectorial step.

5.2 The Stopping Test

As stated above, our algorithm works with successive approximations but the result is obtained with the desired accuracy [1]. That is to say that it is the operator who gives the acceptable tolerance. Suppose that margin tolerance is $\epsilon \in \mathbb{Q}$, with equation notations in 26, then the stopping test is :

$$| \alpha_0 - \alpha_1 | \leq \epsilon \tag{28}$$

5.3 To Provide a Solution

When both $\mathcal{H} \bigcap K \neq \emptyset$ and $| \alpha_0 - \alpha_1 | \leq \epsilon$ are true, a feasible solution is obtained in the part of the polytope as we can see it in Figure 2.

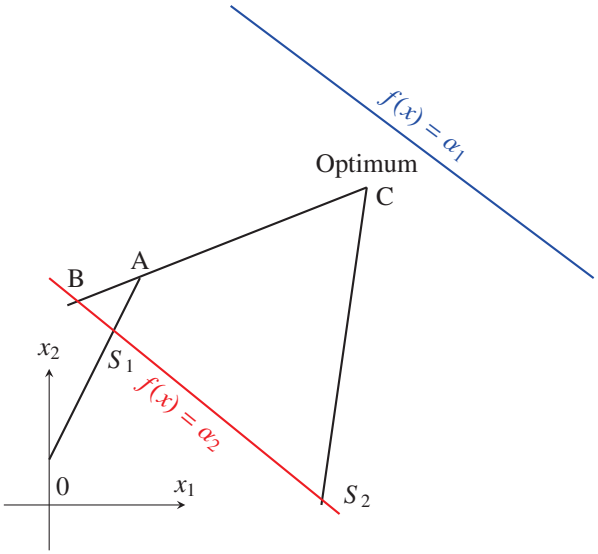

Figure 2. The dichotomic procedure

Geometrical explanation: Unless we have obtained the *optimum*, the hyperplane $\mathcal{H}_0 : f(x) = \alpha$ separate the other constraints. Basis are vertices of the residual polytope. There are n such vertices associated with the separating hyperplane satisfying the constraints. But all interior points of the residual polytope satisfy the constraints and are in the permitted interval. All convex combination of these points is a valid solution[2].

More details about the algorithm can be found in Lavallée, Ndiaye and Seck (2011); and Ndiaye, Lavallée and Seck (2013)

6. Implementation and Numerical Results

In this section we present computational studies on the linearization approaches and the senegaulois algorithm for the TNLU.

[1] It is also the case of the Katchiyan's, Karmarkar's, and Murty's algorithms. And the simplex algorithm is also an approximated one due to the limitation of the coding numbers on computers.

[2] It's true because the set of solutions is convex.

6.1 Implementation

Aiming to obtain the best possible bounds, we solve the linearization model globally with the senegaulois algorithm and the mixed integer model by Cplex (2016) and Gurobi (2016). Simulations are done on a CPU@ 3.40 GHz Intel(R) Core(TM)i7-370 HP-Pro3500 with 16 GB of main memory running Linux ubuntu. The termination criteria, i.e. the difference between the lower and upper bounds in our algorithm, is set to 10^{-5}.

To model the TNLU problem, we use a modeling tool known A Mathematical Programming Language (AMPL) (2018), version 20130609. Developed at Bell Laboratories, AMPL is a high level, an algebraic modeling language for linear and nonlinear optimization problems. Besides its ability to model optimization problems, AMPL also helps in comprehension and completeness of the model logic, due to the similarity of its syntax to algebraic notation. It provides for an easy way to express common linear programming structures (e.g., flow constraints). Although AMPL allows us to mathematically model the optimization problem, we use two external solvers such as Cplex (version 12.5.) and Gurobi to solve the problem.

The entry in AMPL includes a template file (`*.mod`), a data file (`*.dat`), and a runtime program (`*.run`). The AMPL groups them in a format that the solver understands. Our algorithm based on the dichotomous search as an AMPL script, is implemented in the (`*.run`) file. The AMPL execution program calls the script file (`*.run`) which provides the mathematical description of the model and the data into AMPL. Then, it passes this instance of the problem to the solver, which in turn, resolves the instance, and makes the solution to AMPL.

6.2 Numerical Results

In order to access the convergence of the senegaulois algorithm presented in section 5, we have applied it to eight examples. The obtained solutions, with our algorithm, are compared with those obtained with Cplex and Gurobi. The Table 1, with 8 test problems (tnlu1 to tnlu8), shows the result of the computational comparison and the performance of each resolution technics, where:

- `Inst`: is the name of the test problem,
- `NbVar`: the number of variables,
- `OptVal`: the obtained optimal value,
- `Times(s)`: the execution time in seconds,
- `Iter`: the number of iterations,
- α_0: the initial value for the senegaulois algorithm.

We see that the senegaulois algorithm provides good optimal value for the TNLU problem. For some test problems on academic instances, the number of iterations is quite small. For instances tnlu1, tnlu4 and tnlu8 the iterations are quite small compared to Cplex and Gorubi. However, for all instances, the obtained Optimal value remains nearly the same for the 3 methods. The Figures 3 and 4 show the evolution of the number of iterations and computational time of each

Table 1. Optimal values, times(s) and iterations for Cplex, Gurobi and Senegaulois

Inst	NbVar	OptVal			Times(s)			Iter			α_0
		Cplex	Gurobi	Senegaulois	Cplex	Gurobi	Senegaulois	Cplex	Gurobi	Senegaulois	
tnlu1	472	3.4813e+4	3.4813e+4	3.4813e+4	6.072	0.644	0.012	43413	1187	112	3.90003e+4
tnlu2	1265	8.900e+10	8.900e+10	8.900e+10	0.08	0.232	0.056	224	571	341	7.770e+12
tnlu3	2796	9.0392e+10	9.0392e+10	9.0392e+10	0.244	0.26	0.284	826	1566	1010	9.039e+12
tnlu4	9584	3.264600e+6	3.264600e+6	3.264600e+06	13.252	25.484	0.94	281393	205037	1674	6.000e+12
tnlu5	24580	1.71398e+11	1.71398e+11	1.7139e+11	8.488	4.504	7.42	15	15658	4056	1.7139e+12
tnlu6	99932	2.58995e+5	2.58995e+5	2.58995e+5	388.06	17.3	217.448	33008	36079	34759	3.000e+6
tnlu7	173536	2.75102e+5	2.75102e+5	2.75102e+5	738.928	27.14	864.4	24144	62134	66547	3.500e+5
tnlu8	222785	8.68577e+09	8.68577e+09	8.68577e+09	585963	2019.68	235.788	198042112	638826	36102	9.780e+10

algorithm, respectively (Cplex, Senegaulois and Gurobi). From these figures can see that iteration and time frequencies are maintained in senegaulois algorithm.

In addition, to evaluated the results we use the free software R project (2018), and display the 8 observations and the 11 variables:

Table 2. Average of the values.

	NbVar = 66868.75		
methods	Senegaulois	Cplex	Gurobi
Time(s)	165.7935	73389.77	261.9055
Iter	18075.12	24803142	120132.2
α_0	3.077588e+12		

```
'data.frame':  8 obs.  of 11 variables:
$ Nvar:  int 472 1265 2796 9584  24580 99932 173536 222785
$ CplexVal:  num 3.48e+04 8.90e+10 9.04e+10  3.26e+06 1.71e+11 ...
$ GurobiVal:  num 3.48e+04 8.90e+10 9.04e+10 3.26e+06 1.71e+11 ...
$ SenVal:  num 3.48e+04 8.90e+10 9.04e+10 3.26e+06 1.71e+11 ...
$ CplexTime:  num 6.072 0.08 0.244  13.252 8.488 ...
$ GurobiTime:  num 0.644 0.232 0.26 25.484 4.504 ...
$ SenTime:  num 0.012 0.056 0.284 0.94 7.42 ...
$ CplexIter:  int 43413 224 826 281393 15 33008 24144 198042112
$ GurobiIter:  int 1187 571 1566 205037  15658 36079 62134 638826
$ SenIter:  int 112 341 1010 1674 4056 34759 66547 36102
$ SenAlpha0:  num 3.90e+04 7.77e+12 9.04e+12 6.00e+12 1.71e+12 ...
```

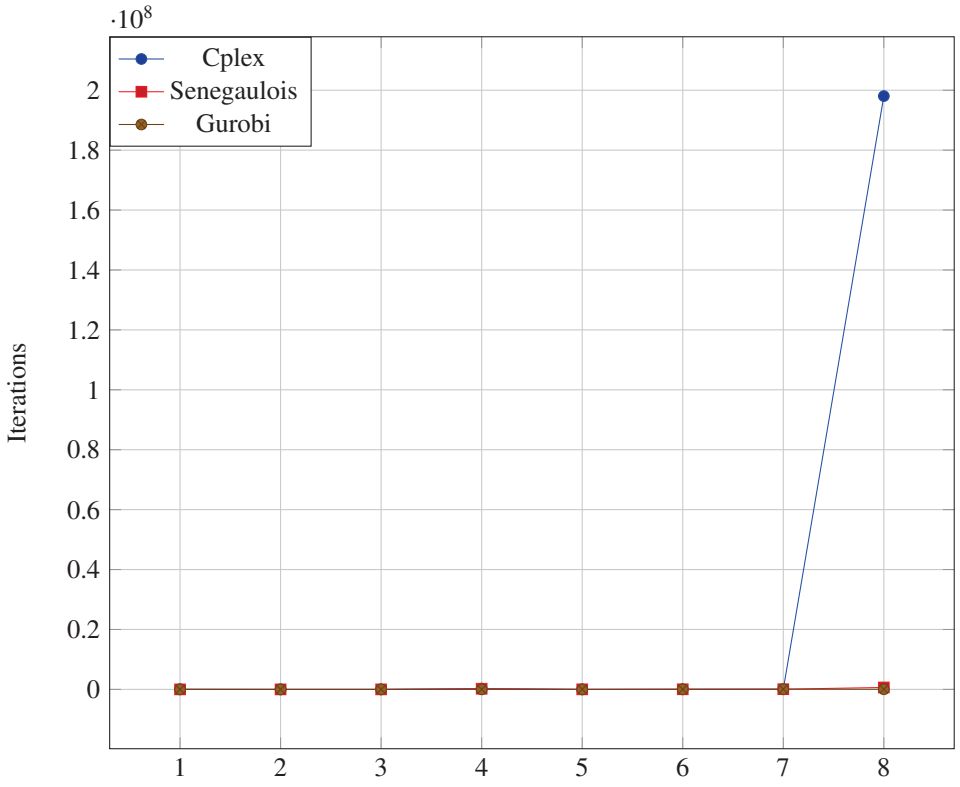

Figure 3. Iterations for the Cplex, Gurobi and Senegaulois

We firt compute only the average of 2 obervations (times and iterations), then the number of variables and the firt value of the senegaulois algorithm.

On average, the same upper bound is appreciably found, but the running time and the number of iterations of Cplex and

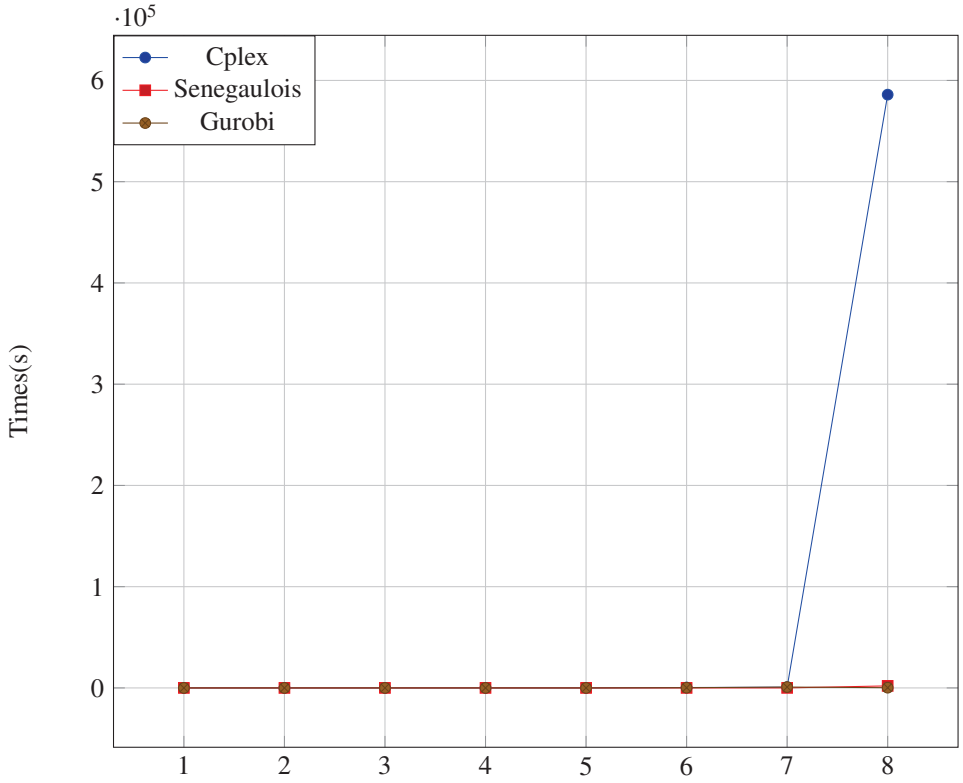

Figure 4. Times(s) for the Cplex, Gurobi and Senegaulois

Gurobi remain large (see Table 2). Our numerical experiments show convergence with a fewer number of iterations and computational times.

7. Conclusion and Future Works

This study aims at describing and simulating a new type of optimization problem for the TNLU problem. Our algorithm for TNLU was presented and applied to problems from the literature designed to highlight some intrinsic difficulties of TNLU. The Senegaulois algorithm, based on a dichotomic search is compared to two programming solvers such as Cplex and Gurobi. We investigated the average of iterations and computational times and established interesting results characterizing some instances. In future work, we plan to study the combination between the senegaulois algorithm and genetic or greedy algorithm.

Acknowledgement

The authors thanks the Non Linear Analysis, Geometry and Applications (NLAGA) Project for supporting this work. The authors thank the anonymous authors for their helpful comments.

References

Achmanov, S. (1984). Programmation linéaire. Edition Mir. Traduit du Russe, édition Russe 1981.

Baldé, M. A. M. T., & Ndiaye, B. M. (2016). A new heuristic method for transportation network and land use problem. *Journal of Mathematics Research, 8*(3), 94-111. https://doi.org/10.5539/jmr.v8n3p94

Belotti, P. (2018). couenne: a user's manual, https://projects.coin-or.org/Couenne, Clemson University.

Billionnet, A., Elloumi, S., & Lambert, A. (2013). An efficient compact quadratic convex reformulation for general integer quadratic programs. *Computational Optimization and Applications, 54*(1), 141-162. https://doi.org/10.1007/s10589-012-9474-y

Blanchard, A., Elloumi, S., Faye, A., & Wicker, N. (2002). Une famille de facettes pour le polytope de l'affectation quadratique. Research Report CEDRIC-02-330, CEDRIC Lab/CNAM.

Boyce, D. E., Farhi, A., & Weischedel, R. (1973). Optimal network problem: A branch-and-bound algorithm. *Environment and Planning, 5*, 519-533. http://dx.doi.org/10.1068/a050519

Brezis, H., Ciarlet, P. G., & Lions, J. L. (1999). Analyse fonctionnelle: théorie et applications, 91, Dunod Paris.

Burkard, R. E., Cela, E., Pardalos, P. M., & Pitsoulis, L. S. (1999). The quadratic assignment problem. In Ding-Zhu Du and PanosM. Pardalos, editors, Handbook of Combinatorial Optimization, Springer US, 1713-1809.

Coelho, L. C. (2018). Linearization of the product of two variables. Retrived from: https://www.leandro-coelho.com/linearization-product-variables/

Duranton, G., & Puga, D. (2015). Urban land use, in: *Handbook of regional and urban economics, 5,* Elsevier, 467-560.

Fortet, R. (1960). L'algebre de boole et ses applications en recherche operationnelle. *Trabajos de Estadistica, 11*(2), 111-118. https://doi.org/10.1007/BF03006558

Fourer, R., Gay, D., & Kernighan, B. (1993). AMPL: A Mathematical Programming Language, 117, Boyd & Fraser Danvers, MA. Retrived from http://www.ampl.com/

Glover, F., & Woolsey, E. (1974). Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations Research, 22*(1), 180-182.

Glover, F. (1975). Improved Linear Integer Programming Formulations of Nonlinear Integer Problems, *22*.

Gueye, S. (2012 January 8). Modèle affectation quadratique / Dimensionnement réseaux projet ortrans.

Gurobi Optimization, LLC (2018). Gurobi Optimizer Reference Manual. Retrived from http://www.gurobi.com

Hoang, H. H. (1973 Jan). A computational approach to the selection of an optimal network. *Management Science, 19*(5). http://dx.doi.org/10.1287/mnsc.19.5.488.

IBM ILOG, *CPLEX Optimization Studio V12.5*, Inc. Using the CPLEXR Callable Library and CPLEX Barrier and Mixed Integer Solver Options. Retrived from http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio.

Kelly, D. J., & O'Neill, G. M. (1991). The minimum cost flow problem and the network simplex solution method. Master's thesis, University College, Dublin 2,4,5,6,11,13,24,27-29,39-46.

Koopmans, T. C., & Beckmann, M. J. (1957). Assignment problems and the location of economic activities. *Econometrica, 13*(52), 53-76.

Lavallée, I., Ndiaye, B. N., & Seck, D. (2011). A new way in linear programming. LISS 2011- Proceedings of the 1st International Conference on Logistics, *Informatics and Service Science, Beijing, China, 8-11 June, 2011*(2), 220-227.

Lawler, E. L. (1963). The quadratic assignment problem. *Management Science, 9*, 586-599. http://dx.doi.org/10.1287/mnsc.9.4.586

Liberti, L., Cafieri, S., & Tarissan, F. (2009). Reformulations in mathematical programming: A computational approach. *Foundations of Computational Intelligence, 3,* 153-234.

Lin, J. J., & Feng, C. M. (2003). A bi-level programming model for the land use-network design problem. *The Annals of Regional Science, 37,* 93-105. http://dx.doi.org/10.1007/s001680200112

Maniezzo, V. (1997). Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. Research Report CSR 97-1, Scienze dell'Informazione, Cesena site, University of Bologna.

Ndiaye, B. M., Lavallée, I., & Seck, D. (2013). Solving worrying simplex's instances in polynomial time. *Mongolian Mathematical Journal, 17,* 5-43.

R Core Team. (2018). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Retrived from https://www.R-project.org

Sahni, S., & Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM (JACM), 23*(3), 555-565. https://doi.org/10.1145/321958.321975

Scott, A. J. (1969). The optimal network problem: Some computational procedures. *Transportation Research, 3,* 201-210. http://dx.doi.org/10.1016/0041-1647(69)90152-X

Watters, L. J. (1967). Letter to the editor-reduction of integer polynomial programming problems to zero-one linear programming problems. *Oper. Res., 15*(6), 1171-1174. https://doi.org/10.1287/opre.15.6.1171

Xia, Y., & Yuan, Y. X. (2006). A new linearization method for quadratic assignment problem. *Optimization Methods and Software, 21*(5), 803-816. http://dx.doi.org/10.1080/10556780500273077

Zhang, H., Beltran-Royo, C., & Ma, L. (2010). Solving the quadratic assignment problem by means of general purpose mixed integer linear programming solvers.