A brief introduction to LENSOLVER as linear programming application

N A Pratomoatmojo
Urban and Regional Planning Department, Institut Teknologi Sepuluh Nopember
Email: pratomoatmojo@urplan.its.ac.id

Abstract. Linear programming technique is a model that can be used to solve the problem of allocating resources with limited conditions. This technique is often used to solve problems in several good fields. Linear Programming is a deterministic tool in which all model parameters are assumed to be known with certainty. In addition, the use of linear programs can be used to make decisions from various alternative objectives in the form of activities or strategies, where there are constraint functions considered. In this paper introduces LENSOLVER, as an application of linear programming application. In this early stage of development, LENSOLVER is developed based on the PuLP framework. PuLP is a library for the Python programming language that allows users to create mathematical programs for problem-solving optimization and decision making by means of calculating variables using mathematical expressions. However, this first development of LENSOLVER has a limitation of not supporting non-linear optimization. Considering the ease and reliability LENSOLVER was developed to reach non-programming users who want to implement various linear programming techniques in several cases in operations research.

1. Introduction
Linear programming technique is a model that can be used to solve the problem of allocating resources with limited conditions. It is also known as an analytical planning technique in which the analysis uses mathematical methods to solve the problem of allocating resources and limited products to obtain optimal benefits. A linear program is mathematical programming in which there are objective functions in the form of unknown linear and constraint functions in the form of linear equations or linear inequalities (Kakiay, 2008). The objective of the Linear Program is an outcome that achieves the stated goals (optimal) in the best possible way among all possible alternatives within the limits of available resources (Dumairy, 2012). This method is very helpful in solving a problem in the form and arrangement of the presentation of problems that will be solved by linear programming techniques. This technique is often used to solve problems in several fields. According to Taha (1975), it is mentioned that Linear Programming is a deterministic tool in which all model parameters are assumed to be known with certainty. In addition, the use of linear programs can be used to make decisions from various alternative objectives in the form of activities or strategies, where there are constraint functions considered. The constraint function and objective function are written in the form of equations and inequalities in the decision variables (Agustini and Rahmadi, 2004).
In linear programming model, there are 2 kinds of functions, namely objective function and constraint functions. The objective function is a function that describes the goal/target in linear programming problems related to the optimal management of resource to obtain maximum benefits or minimal costs (Supranto, 1991), meanwhile the goal to be achieved must be realized in a linear mathematical function. The constraint functions is a set of variation of variables function due to each limitation of problem-solving. Furthermore, the function is maximized or minimized against existing constraints. The management department will face various obstacles in realizing its goals which are constraints on the collection of decisions that might be made and must be put into linear mathematical functions.

LENSOLVER is a GUI linear programming application developed based on a PuLP model framework. PuLP is a free library for the Python programming language that allows users to create mathematical programs in which there are facility solutions to optimization problems and decision variables represented as close as possible to original mathematical expressions (Mitchel et.al., 2011). PuLP supports various forms of software development (solver software) both for commercial, open-source solver or integrated as development. Since LENSOLVER developed based on PuLP so it has a limitation of not supporting non-linear optimization but supporting linear and mixed-integer models. Considering the ease and reliability of PuLP in handling various linear optimization problems, LENSOLVER was developed to reach non-programming users who want to implement various linear programming techniques in various cases in operations research. The GUI on LENSOLVER is designed and developed to make it easier for users of existing optimization software, especially commercial applications, such as Lingo, Excel Solver, etc.

2. Syntax and Optimization Concept

The algorithm used to process optimization operations on the LENSOLVER-based algorithm developed in PuLP, the Revised Simplex Method or Interior Point Methods. At the present state, LENSOLVER has several optimization concepts, namely Linear Programming (LP), Integer Programming (IP), and Mixed-Integer Linear Programming (MILP).

LENSOLVER is designed for ease to write the syntax for non-programming users and also makes it easier for users who are familiar with other commercial applications such as Lingo. Even writing is considered far easier than Lingo for solving linear optimization problems. But to do scripting in LENSOLVER, there are some rules for writing equations that must be followed in order to run smoothly. Several features definition in LENSOLVER:

- **NUMVAR** is the number of variables involved both in its definition and contained in the constraint function (Constraint Function - CF). NUMVAR is filled in with the number of variables to be analyzed. Writing the number of variables in the form of integers 1 and so on
- **VAR** is the variable to be defined in the equation. The writing of variables in LENSOLVER version 1.0 is limited between X1, X2, to Xn.
- **LPSOL** (Linear Programming Solution) is a solution for linear programming, there are 2 types namely "Maximize" (operations to maximize object-function) and “Minimize” (operations to minimize the value of object-functions)
- **OF** (Object Function) is an objective function that will be optimized and it leads to answer the questions of optimization.
- **TYPEOPTVAR** (Type Optimized Variable) is the type of format for the optimized variable; “Integer” or “Float”
- **NUMCONS** is the number of constraint functions involved in the optimization process, for example: 2, if there are 2 constraint functions that will be processed
- **CF** (Constraint Function) is a set of function that is consist of all limitation function to reach the goal. The function must clear and operable. In between variables and coefficient must be followed by * as a representation of multiplication.
Writing a constraint function, the smaller value has to be written first rather than putting the larger value on the statement of function.

$$330 \leq X_1 \leq 1000 \ (Right)$$

$$1000 \geq X_1 \geq 300 \ (Wrong)$$

In general the formulas used for linear programs are as follows;

$$f(X) = \sum_{i=1}^{n} C_i \cdot X_i \quad \text{Objective Function (max or min)}$$

$$X = (X_1, X_2, X_3, ..., X_n) \quad \text{Variables}$$

$$\sum_{i=1}^{n} a_{ij} \cdot X_i \leq b_j, i = 1, 2, ..., n \quad \text{Constraint Functions}$$

$$X_i \geq 0, i = 1, 2, ..., n$$

3. Example Case and Solution
This case will be within the topic of optimization of land use composition using hypothetical data. The city of Kotalama has four land use classes, i.e. settlement of workers, commercial and business area, industry and vacant land. Kotalama area is 2,200 Ha, and it lies settlement 100 Ha, Industry 200 Ha, commercial area 150 Ha, and the rest is bare-land. The capacity of the settlement area is about 100 population/Ha which only serves the worker of City Kotalama. The worker that can be hired for industry and commercial area is 120 people/Ha and 200 workers/Ha respectively. In the City of Kotalama, the tax is used to generate income for the city based on the area of each class of land use. The tax for the settlement area as worker housings is Rp. 55,000,000/Ha, the industry Rp. 60,000,000/Ha and the commercial area is Rp. 70,000,000/Ha. It also informed that the maximum carrying capacity for the population (workers) in Kotalama is 120,000 inhabitants for the entire city. The question is What is the optimal area of each land use that can still be developed to get the most profit through taxes?

Solutions:

If Settlements area is X1, Industry is X2 and commercial area is X3;
Based on the questions:

**Solution: find the maximum profit**

$$55,000,000 \cdot X_1 + 60,000,000 \cdot X_2 + 70,000,000 \cdot X_3$$

**Constraints in mathematic expression**

The total area should less than 2,200 Ha:

$$X_1 + X_2 + X_3 \leq 2200$$

Max. population (worker only) should less than 120,000 inhabitants:

$$120 \cdot X_2 + 200 \cdot X_3 \leq 120000$$

Total number of workers (Commercial and Industry) should be all accomodated in settlement area:

$$120 \cdot X_2 + 200 \cdot X_3 \leq 100 \cdot X_1$$

Existing for each of land use (already developed), indicate not possible for less than existing:

$$X_1 \geq 100$$

$$X_2 \geq 200$$

$$X_3 \geq 150$$
These are several stages of sequences to perform linear programming using LENSOLVER:

1. Determine the number of variables involved. The case expresses 3 variables that will be taken into account for optimization, i.e. settlement area, commercial area, and industrial area.

2. Define the variables, in this case, X1, X2, and X3 will be used as variables. The number of variables must be the same as the variables determined at the initial stage, which is 3. These variables are referred to as Decision Variables.

3. Define the solution that will be processed, Maximization or Minimization. It depends on the optimization question. It often the real-world problems tends to be formulated as minimizations instead of maximization since planners always seem to be a pessimist (Vanderbei, 2008).

4. Define the type of optimization process, is the solution will be on integer value or float. The integer will generate a solution for each variable in integer format, while “Float” possible to optimize the variables in more precisely, and possible to get the decimal values for each variable as a solution.

5. Determine the number of constraints (NUMCONS) as an integer number. In this case, 5 represents constraint function or 5 lines of syntax constraints.

6. Define the constraint factors (CF). Per line should be filled by one constraint function.

Figure 1 shows the formulation filled in LANSOLVER application, while Figure 2 expresses the result of the optimization solution. The optimal solution for maximizing profit in detail, as follows;

| Land-use    | Pop/Ha | Optimized (Ha) | Pop   | Tax            | Profit              |
|-------------|-------|---------------|-------|----------------|---------------------|
| Settlement  | 100   | 1520          | 152,000 | 55,000,000 | 83,600,000,000      |
| Industry    | 200   | 200           | 24,000  | 60,000,000  | 12,000,000,000      |
| Commercial  | 150   | 480           | 9,600   | 70,000,000  | 33,600,000,000      |
|             | 2,200 |               |        |               | 129,200,000,000     |
| Total Worker (Industry + Commercial) | 120,000 |               |       |               |                     |
| Total Worker < Settlement Capacity | True (120,000 < 152,000) |               |       |               |                     |

Table 1 explains the validation check on the stylesheet. The model of optimization for three types of land use not exceed 2,200 Ha which is consists of 1,520 Ha of the settlement area, 200Ha of industry, and 480Ha of commercial area. The population also not exceed 120,000 inhabitants as well as less than settlement capacity (152,000 inhabitants). The maximum profit for the optimal composition is Rp 129,200,000,000.

In addition, LENSOLVER also prove able to solve a complex constraint function. According to the example explained above, the solver able to calculate variables on the left side and the right side of inequality function at the same time (it refers to constraint function: 120*X2 + 200*X3 <= 100*X1). The computation succeeds to optimize the value of X2, X3, and X1.
Figure 1. LENSOLVER Graphic User Interface

Figure 2. LENSOLVER’s report console
4. Conclusion
This paper is intended to demonstrate LENSOLVER as an application that capable to perform optimal optimization operations by means of the linear programming technique, and it also proved that LENSOLVER able to solve the linear problem. At this early stage of development, LENSOLVER is still succeeding in implementing the framework of PuLP, where the limitation of PuLP itself is that it cannot carry out non-linear optimization analysis. This application is considered capable of providing the functions and outputs needed by many linear optimization users, both academics and researchers. It also believed PuLP still has many features for optimization problems. Regarding its potential development and wide-users, LENSOLVER in the future will be developed using other frameworks such as Pyomo (Python Optimization Modeling Objects) to improve the function and scalability of its use.

The optimization case in terms of land use composition under the urban planning context shows it has a great potential impact in the technical planning process. However, the goals still lack spatial information in geographical information systems (GIS). Therefore, the combination of LP model with other spatial-simulation tools such as Metronamica (2005), CLUE (Verburg and Overmars, 2009), and LanduseSim (Pratomoatmojo, 2014) will offer a robust solution in estimating an optimal land use composition, spatial distribution, and land use change in mapping perspective.

5. References
[1.] Agustini, D.H. dan Rahmadi, Y.E. (2004). Riset Operasional Konsep-Konsep Dasar.
[2.] Dumairy (2012) Matematika Terapan untuk Bisnis dan Ekonomi edisi Kedua. BPFE
[3.] Kakiay, T.J. (2008) Pemrograman Linier. Andi, Yogyakarta
[4.] Mitchel S., O’Sullivan, M., Dunning, I. (2011) PuLP: A Linear Programming Toolkit for Python. Download at: http://www.optimization-online.org/DB_FILE/2011/09/3178.pdf
[5.] Pratomoatmojo, N.A (2014) LanduseSim sebagai aplikasi pemodelan dan simulasi spasial perubahan penggunaan lahan berbasis Sistem Informasi Geografis dalam konteks perencanaan wilayah dan kota. CITIES international conference.
[6.] Supranto J. (1983) Linear Programming. Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia
[7.] Taha H.A. (1975) Integer Programming – Theory, Applications, and Computations. Academic Press.
[8.] Vanderbei R.J. (2008) Linear Programming: Fondations and Extensions, Third Edition. Springer
[9.] Verburg P.H. and Overmars, K.P. (2009) Combining top-down and bottom-up dynamics in land use modeling: exploring the future of abandoned farmlands in Europe with the Dyna-CLUE model. Landscape Ecology 24 (2009)
[10.] Van Delden, H., Escudero, J.C., Uljee, I., Engelen, G. (2005) METRONAMICA: a dynamic spatial land use model applied to Victoria-Gasteiz. Virtual Seminar of the MILES Project. Centro de Estudios Ambientales, Vitoria -Gasteiz.