

## Optimal Algorithm Design Of Center of Graph Based on IBM ILOG OPL Model

Lixin ZHAO<sup>1, a</sup>, Lishuan HU<sup>1, b</sup>

<sup>1</sup>Department of logistics, college of automation, Beijing Union University, Beijing, 100101, China

<sup>a</sup>zdhtlixin@buu.edu.cn, <sup>b</sup>zdhtlishuan@buu.edu.cn

**Keywords:** OPL, Shortest-path Problem, Center of Graph

**Abstract.** In this paper, Optimization problem is introduced. Some computer tool-kits used to solve optimization problems are outlined with their respective strength and weakness. A state of the art optimization programming language-IBM ILOG OPL is presented. This high level OPL facilitates modeling and solving optimal problem in all application fields significantly. An IBM ILOG OPL model of the shortest-path problem is implemented. Then, with IBM ILOG OPL model of Shortest-path problem as building block, an algorithm of center of graph is illustrated through calling this building block but with different data in consecutive way. This algorithm can be implemented with IBM ILOG Script language. This approach enable engineers in all kinds of field to solve optimal problems since no underlying mechanism of optimization need to be grasped.

### Introduction

Optimization algorithms aim to get the solution of some optimization problems that user want by applying some rules and approaches into searching process based on some mechanism or principles. Optimization problem involves decision-making problem, including designing and planning problem, decision analysis problem, controlling and optimization problem in such fields as engineering, social economy, scientific management, industry&agriculture production and military[1], which usually fall into one of the following branch of OR: linear programming, integer programming, dynamic programming, graph and network analysis and game theory etc if solved by classical algorithm. With the rapid development of computer software technology, some specialized computer tool-kits are developed to solve optimal problems without the need to design an algorithm then implement it with some programming language such as C, C++, Java, FORTRAN etc. For instance, WinQSB and ORS. On one hand, those specialized tool-kits simplify modeling, solving and evaluating process significantly. On the other hand, data need to be tailored to satisfy the requirement of the software tool-kits, resulting in less flexibility.

Another computerized approach used to solve optimization problem is high level optimization programming language, including Lindo/Lingo, AMPL, Matlab and IBM ILOG OPL etc. Unique syntax and semantics, high ability of model expression are their outstanding features.

IBM ILOG OPL(optimization programming language)has been used to solve the optimization problems for many Fortune 500 enterprises. A large number of universities globally are using the solution from IBM ILOG OPL[2]. IBM ILOG OPL is endowed with modeling language corresponding to the notion of advanced algebra and set, as well as constraint language for constraint programming. An IDE(integrated development environment) was designed to integrated an editor, compiler, solver, debugger and ODM. IBM ILOG OPL isolates data from model file, which contribute to its reusability. IBM ILOG Script language goes with IBM ILOG OPL, so to make the model run in a consecutive way, during which data can be modify, comparison can be made in the model, which qualify IBM ILOG OPL for the complicated tasks that can not be done with other optimization programming languages.

## Language Features

IBM ILOG OPL is endowed with unique syntax and semantics, apart from absorbing the essences of many programming languages, including tuple, associate array, set, logic constraint and loop structure. Its statements are concise and gorgeous, which is proved in expressing complicated mathematical model.

Data type of ILOG OPL includes int, float, string, Boolean, of which, int, float and Boolean can be used to declare decision variable. Another two decision variable types are interval and sequence in constraint programming. Meanwhile, IBM ILOG OPL has complicated data structure, such as range, array, set and tuple[3]. These data structure qualify IBM ILOG OPL a powerful language in solving complicated problem, in which, the logical structure and physical structure of data is complicated. More importantly, generic array, generic set and associate array equip IBM ILOG OPL with powerful capacity to transform data to one form to another to meet the need of the model.

IBM ILOG OPL implements self-defined type-tuple to define single data of tuple, set of tuple, and array of tuple. The ideology of tuple is analogous to tuple of relational database, which is counterpart of entity of conceptual model, relation of logical model and table of physical model. Programmers can manipulate data of tuple type as much as records in relational database.

IBM ILOG not only implement OPL language, but IBM ILOG script language, which is mainly used to data preprocessing, data post-processing and flow control. Within data preprocessing, data post-processing and flow control, IBM ILOG Script language is an implementation of JavaScript. IBM ILOG Script language can interact with IBM ILOG OPL during execution, including modify model data structure, auxiliary data structure, logic judgment etc, which enable IBM ILOG OPL to solve complicated optimal problems in a more extensive sense in a totally different perspective.

## Application

### 1) Shortest-path Problem

A manufacturing concern has a warehouse at location X and the market for the product at another location Y. Given the various routes of transporting the product from X to Y and the cost of operating them, what is the most economical way of transporting the products? This problem comes under a type of graph theoretic problem known as “shortest-path problem”[4]. This problem can only be solve by designing an algorithm and implement it with a general programming language, such as C, C++ or Java. Relaxation algorithm is used to solve all types of single source Shortest path Problem. Dijkstra algorithm is the most efficient algorithm used to solve single source Shortest path Problem with non-negative weight. Floyd’s algorithm is used to solve shortest path of all pairs of sources in a network.

Let us take a look at the way that IBM ILOG OPL solve it.

For a single connected weighted graph  $G=(V,E)$  with  $n$  vertex in the vertex set  $V$ , with  $e$  edges in the edge set  $E$ , let  $w_{ij}$  ( $i, j \in V$ ) be the weight matrix of  $G$ . Let  $s$  and  $t$  be the specified two vertices between which the shortest path will be solved.

Suppose:

$x_{ij}$  denote the decision that whether the edge from vertex  $i$  to vertex  $j$  should be on the shortest path.

If this edge is one of the edge of shortest path,

$x_{ij}=1$

else

$x_{ij}=0$

Firstly, mathematical model used to solve the shortest path between  $s$  and  $t$  is as such

$$\min \sum_{(i,j) \in E} w_{ij} x_{ij} \quad (1)$$

$$s.t. \quad \sum_{\substack{j=1 \\ (i,j) \in E}}^n x_{ij} - \sum_{\substack{j=1 \\ (j,i) \in E}}^n x_{ji} = \begin{cases} 1, i = s \\ -1, i = t \\ 0, i \neq s, t \end{cases} \quad (2)$$

$$x_{ij} = 0 \text{ or } 1, (i, j) \in E \quad (3)$$

Secondly, this mathematical model is implemented through IBM ILOG OPL.

The data structure used is adjacent matrix named Adjmatrix. Then Adjmatrix is transformed into a Adjacent table named Adj, tuple structure of which is as such:

```
tuple adj
{
key int origin;
key int destination;
int weight;
}
{adj} Adj={<i,j,w>|i in 1..n,j in 1..n:adjmatrix[i][j]!=0};
```

The origin point set, destination point set and intermediate point set are declared as orginpoint, destinationpoint, and midnodes.

Decision variable is a Boolean array with Adj as index.

```
dvar boolean x[Adj];
```

Objective function is to minimize the summation of edge weight in the shortest path.

```
minimize sum(i in Adj)x[i]*i.weight;
```

The constraints that decision variables are subject to include:

```
sum(i in Adj:i.origin in orginpoint)x[i]==1;
```

```
sum(i in Adj:i.destination in destinationpoint)x[i]==1;
```

```
forall(i in midnodes)
```

```
sum(<j,l,w> in Adj:l==i) x[<j,l,w>]==sum(<l,k,w> in Adj:l==i)x[<l,k,w>];
```

This shortest-path model can only solve a shortest path between two specified vertices. In order to get a set of shortest path between all pairs of vertices of G, the data of origin point set and destination point set will be modified manually in consecutively way.

## 2)Center of Graph

The center (or Jordan center[5]) of a graph is the set of all vertices of minimum eccentricity[6]. that is, the set of all vertices A where the greatest distance  $d(A,B)$  to other vertices B is minimal. Equivalently, it is the set of vertices with eccentricity equal to the graph's radius[7]. Thus vertices in the center (central points) minimize the maximal distance from other points in the graph.

In supply chain network design based on service level, the goal to locate a facility is to meet the needs of the customers to the greatest extent. In order to serve all the customers as good as possible, one of the goals is to minimize the worst-case distance to the facility. For example, placing a distribution center at a central point of supply chain network to reduces the longest distance that customers are to the distribution center.

Likewise, an algorithm based on shortest-path model of IBM ILOG OPL can be used to solved center of graph problem in which shortest-path model of IBM ILOG OPL will be solved consecutively.

To solve this problem, the methodology is as followed.

1)Get all the shortest path of every source.

2)Get the shortest path among the longest path of original point to other points.

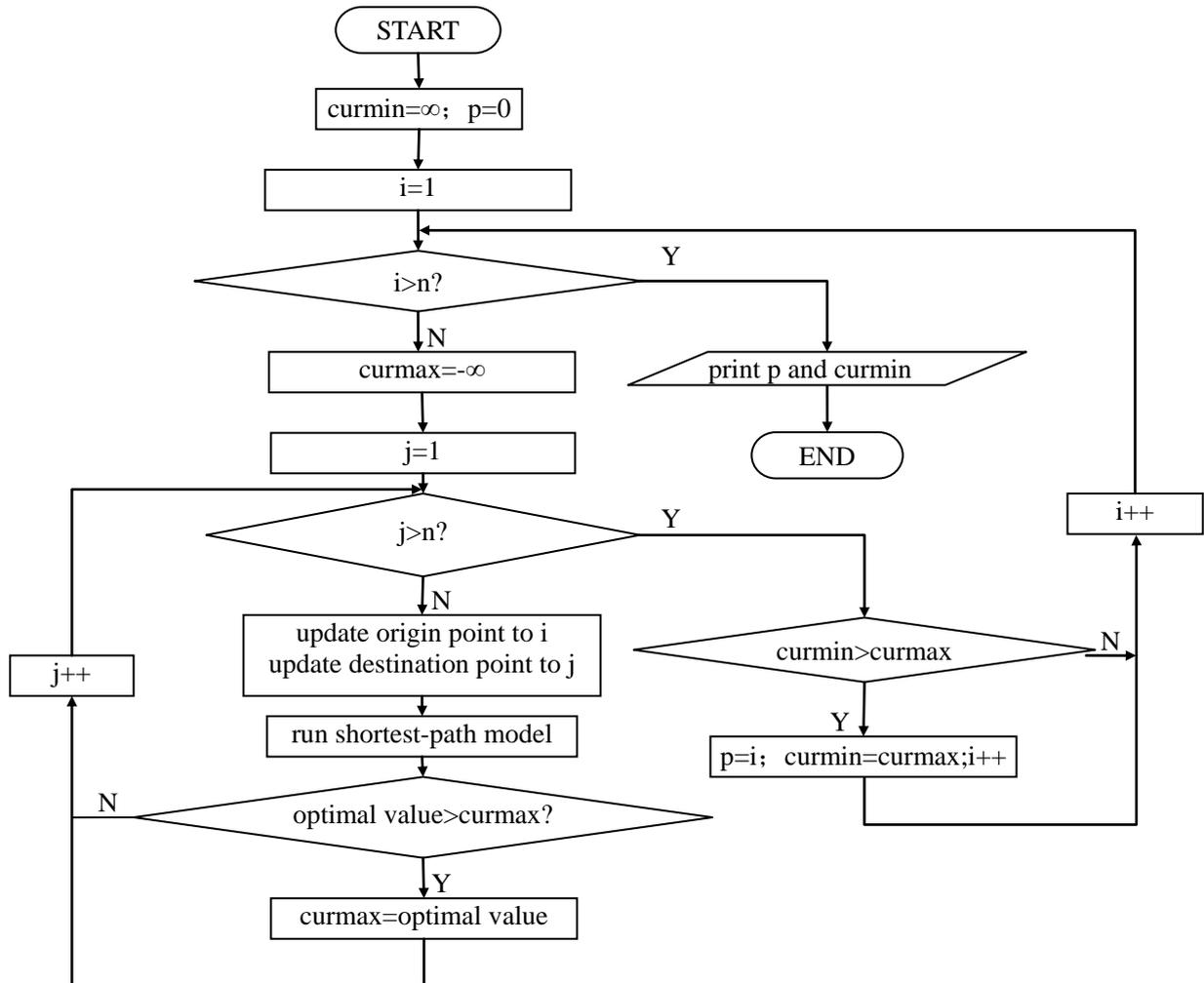
Flowchart of this algorithm is showed in graph 1. The value of p is the center of graph and the value of curmin is the minimized worst-case distance between customers and the center of graph p. This center of graph algorithm must be implemented with IBM ILOG Script language within the same IBM ILOG project and data must be arrange in isolated way, so flow control of Script language can interact with model data so to supply different data with model in each loop.

### 3)Complexity

The complexity of this algorithm is unknown since the complexity of Shortest-path model is unknown.

### Conclusion

IBM ILOG Optimization Programming Language and IBM ILOG Script Language facilitate solving optimal problem with computer. All kinds of engineers can solve the optimal problems in their field without the need to know the underlying mechanism of optimization and to learn a general programming language. More importantly, IBM ILOG OPL model has strong adaptability, which enable it to be extended to relevant application with ease.



Graph 1 algorithm flowchart

### Acknowledgment

This paper is sponsored by “Ministry of Education-IBM Specialty Comprehensive Reform -OR” and “Foreign Visiting Scholar of Beijing Union University” program.

### References

- [1]Baocong JIAO,Lanping CHEN, Methodology and method of of Operations research and its application, Peking University Press,2007.9
- [2]Digital optimization of industry standard CPLEX ILOG, Operational Research and Management[J], 2009.2

[3] Optimization modeling with IBM ILOG OPL workbook

[4] R. Balakrishnan, K. Ranganathan, A textbook of graph theory, Springer press, 1999

[5] Wasserman, Stanley, and Faust, Katherine, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994

[6] McHugh, James A., Algorithmic Graph Theory, Prentice Hall, 1989

[7] Weisstein, Eric W., "Graph center", MathWorld.