CORNERSTONE
MINNESOTA STATE UNIVERSITY MANKATO

Journal of Undergraduate Research
at Minnesota State University,
Mankato

Volume 4 Article 10

2004

Comparison of Optimization Techniques in Large Scale Transportation Problems

Tapojit Kumar
Minnesota State University, Mankato

Student Agreement:
I am submitting my research article to be published in the JUR (The Journal of Undergraduate Research at Minnesota State University, Mankato), an electronic journal of the Minnesota State University Undergraduate Research Center.

I/We certify have followed the accepted standards of scientific, creative, and academic honesty and ethics.

I understand that my article submission will be blind-reviewed by faculty reviewers who will recommend acceptance for publication; acceptance with revisions; or reject for publication.

I understand that as author, I retain the right to present any part of the research in any form in other publications.

The JUR has the right to reproduce and reprint published submissions for instructional or promotional purposes.

For complete details, see *Journal of Undergraduate Research at Minnesota State University, Mankato policies page.*

Mentor Agreement:
I have reviewed the submission, and I support its inclusion in the JUR (The Journal of Undergraduate Research at Minnesota State University, Mankato). I understand that I will be acknowledged as the faculty mentor for the student author(s). To the best of my knowledge, the student has followed the accepted standards of scientific, creative, and academic honesty and ethics.

COMPARISON OF OPTIMIZATION TECHNIQUES IN LARGE SCALE TRANSPORTATION PROBLEMS

Tapojit Kumar (Computer and Information Sciences)
Susan M. Schilling, Faculty Mentor (Computer and Information Sciences)

Abstract

The Transportation Problem is a classic Operations Research problem where the objective is to determine the schedule for transporting goods from source to destination in a way that minimizes the shipping cost while satisfying supply and demand constraints. Although it can be solved as a Linear Programming problem, other methods exist. Linear Programming makes use of the Simplex Method, an algorithm invented to solve a linear program by progressing from one extreme point of the feasible polyhedron to an adjacent one. The algorithm contains tactics like pricing and pivoting. For a Transportation Problem, a simplified version of the regular Simplex Method can be used, known as the Transportation Simplex Method. This paper will discuss the functionality of both of these algorithms, and compare their run-time and optimized values with a heuristic method called the Genetic Algorithm. Genetic Algorithms, pioneered by John Holland, are algorithms that use mechanisms similar to those of natural evolution to encourage the survival of the best intermediate solutions. The objective of the study was to find out how these algorithms behave in terms of accuracy and speed when a large-scale problem is being solved.

Transportation Problem

The Transportation Problem is a classic Operations Research problem where the objective is to determine the schedule for transporting goods from source to destination in a way that minimizes the shipping cost while satisfying supply and demand constraints.

A typical Transportation Problem has the following elements:

1. Source(s)
2. Destination(s)
3. Weighted edge(s)

The objective of a Transportation Problem is to find a minimal cost path from source nodes to destination nodes and meet supply and demand constraints.

A simple transportation network is shown in figure 1. Factories 1 and 2 are the source nodes, warehouses 1, 2 and 3 are the destination nodes, the arcs between nodes represent the existence of a path and the numbers on the arcs represent the cost of shipping each unit product through that specific path. For example, from factory 1 to warehouse 2, the cost of shipping is 6 dollars per unit. The supply and demand requirements appear beside the nodes. For example, factory 1 is capable of producing 30 units and warehouse 1 needs at least 10 units.

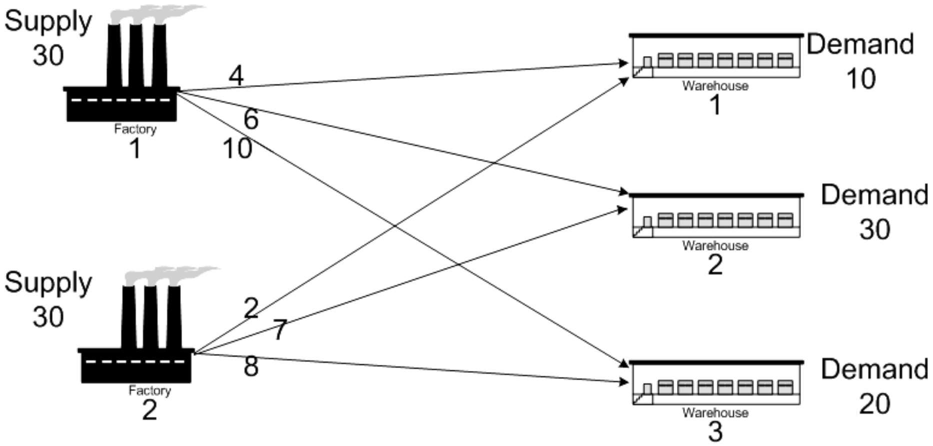

Figure 1

All the information can also be stored in a tableau form. Figure 2 shows the tableau form of this particular example.

	Dest 1	Dest 2	Dest 3	Supply
Source 1	4	6	10	30
Source 2	2	7	8	30
Demand	10	30	20	

Figure 2

This problem is simple and can be solved without much effort. However, practical transportation networks are much more complicated and intelligent techniques are needed to solve them efficiently.

One way to solve Transportation Problems is through a method called Linear Programming. The next section will briefly discuss how this can be done.

Linear Programming

Linear Programming is the mechanism of maximizing or minimizing a linear function over a convex polyhedron. Linear Programming applies to optimization problems in which the objective and constraint functions are strictly linear. The technique is used extensively in a variety of applications, and in areas such as economics, transportation, production, and social sciences [3].

Model

A Linear Programming (LP) problem has the following elements:

1. Variables
2. Objective function
3. Constraints

Variables

In a LP problem, certain quantities are measured. These quantities are represented by variables. For example, in a production problem, a variable, X_1 can represent number of units produced by a source.

Objective Function

The objective function is an equation that represents the goal of the problem.

MAXIMIZE $Z = 4x + 5y$ is an objective function. It gives us the information that this is a maximization problem.

Constraints

Constraints are the set of equations and/or inequalities that restrict the solution space of the problem. If a problem is not constrained by equations, the solution space will not be well defined.

Example

```
1) Max Z = 4X + 5Y
2) Subject To:
3) X+Y=40
4) Y<20
5) X,Y >= 0
```

In this example, line 1 contains the objective function. Line 2 indicates that below are the set of constraints. Lines 3, 4 and 5 contain all the constraints for this problem.

General Solution Method

Linear Programming problems can be solved either graphically or algebraically. However, a problem with more than three dimensions is cumbersome to graph. In general, LP problems are solved by a technique called the Simplex Method, that will be discussed later in this paper.

A graphical method of solving an LP problem has the following steps:

1. **Determination of the Feasible Solution Space:**

First, the non-negativity constraints are accounted for. In figure 3, the horizontal axis X and the vertical axis Y represent the exterior and interior- point variables, respectively. Therefore, it restricts the solution space to the first quadrant.

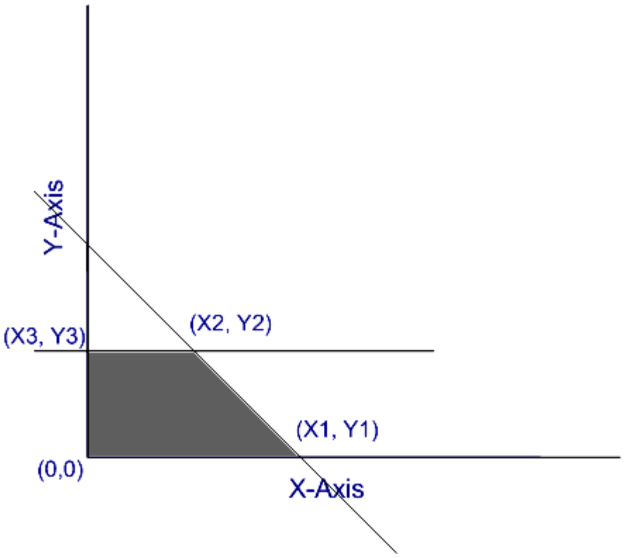

Figure 3

To account for the remaining constraints, first, the inequality is removed by an equation and then the equation is graphed. After all the constraints are accounted for, a solution space is located and is shaded. This solution space defined by the constraints is called the set of feasible solutions.

2. Determination of the Optimum Solution:

An optimum solution exists at a corner point of the solution space. In figure 3, the corresponding corner points are (0,0), (X1, Y1), (X2, Y2), and (X3, Y3). To find the optimum value, first, all these corner points are measured and the values are inserted in the objective function. Then, the obtained values from the objective function are compared and the highest value (since it is a maximization problem) is determined to be the optimum solution.

Solving a Transportation Problem as a LP

Transportation Problems (TSP) can be modeled as a Linear Programming problem. To set up the Transportation Problem as a LP problem, the following elements need to be considered:

Variables

The variables in the LP model of the TSP will hold the values for the number of units shipped from one source to a destination. A variable with double subscripts is used for this problem.

X_{ij} = Number of units shipped from Source i to Destination j

Objective Function

The objective function contains costs associated with each of the variables. It is a minimization problem.

Let C_{ij} denote the cost of shipping one unit from Source i to Destination j.

$$\text{Minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$$

Constraints

The constraints are the conditions that force supply and demand needs to be satisfied. In a Transportation Problem, there is one constraint for each node.

Let S_i denote source capacity and D_j denote destination needs.

$$\sum_{j=1}^{n} X_{ij} \leq S_i \text{ for } i = 1, 2, \ldots, m$$

$$\sum_{i=1}^{m} X_{ij} \geq D_j \text{ for } j = 1, 2, \ldots, n$$

$X_{ij} \geq 0$ for all i and j

Simplex Method

The most general technique of Linear Programming is called the Simplex Method, developed by George B. Dantzig in 1947 [1]. Since that time is has been further developed as a rapid computational technique. Several versions of the Simplex Method now exist.

The Simplex algorithm solves a LP problem algebraically. It uses a tableau form of representing the numbers. The algorithm has two basic parts. First, it finds out whether a given basic feasible solution (BFS)[1] is an optimal solution. If not, it obtains an adjacent BFS with a larger or smaller value (depending on whether the problem is maximization or minimization) for the objective function. The Simplex Method is a greedy algorithm. It obtains basic feasible solutions by making the most improvement from the previous solution in every iteration.

Basic Feasible Solutions

There are various methods to find basic feasible solutions for Transportation Problems. Some of them are Northwest Corner Method, Vogel's Method, Minimum-Cost Method, and MODI Method [7]. Usage of these methods is discussed later in this paper.

Transportation Simplex Method

The Transportation Simplex Method is a special version of the Simplex Method used to solve Transportation Problems. Although it has the basic steps as Simplex Method, it has a much more compact tableau form. This compact form takes less memory, therefore might be faster.

For detailed steps of the Simplex and Transportation Simplex Method, see [7].

Computational Difficulties

Any Linear Programming problem can be solved by using the Simplex Algorithm. With the Transportation Simplex Method, Transportation Problems can be solved accurately. However, it requires too much computation time to solve large-scale problems with these methods. Transportation Problems are Integer Linear Programming problems in essence. When the Simplex Method is used, a lot of intermediate points are found by the algorithm that eventually get rejected. Therefore, it takes too long for these algorithms to solve TSPs.

Since there are special problems where existing generic algorithms are not efficient, heuristic methods exist. Generally, heuristic methods are approximation algorithms that do not necessarily give the optimal solution, but are much faster.

[1] A Basic Feasible Solution (BFS) is an intermediate solution that is feasible, but not necessarily optimum.

Heuristic Method

In this research project, a heuristic method to solve Transportation Problems was developed by means of Genetic Algorithms.

Genetic Algorithms

Genetic Algorithms, pioneered by John Holland, are algorithms that use mechanisms similar to those of natural evolution to encourage the survival of the best intermediate solutions [6]. In Genetic Algorithms, strings of bits are created in a structured way to form a search algorithm for a given problem [2]. Genetic Algorithms (GA) use the concept of survival of the fittest by rejecting strings that are not good enough to take part in producing the next generation. In every generation, a new set of strings is created using parts of the fittest strings from the previous generation(s). Therefore, creation of these strings is not simply randomized. Generations of strings hold characteristics of previous generations. Since survival of the fittest is enforced, strings become better with each generation. After creation of a few generations, the best value is selected as the optimal solution [5].

The three basic operations of Genetic Algorithms are:

1. Reproduction- creation of new generations
2. Crossover- interchanging of parts of parent strings into the child string
3. Mutation- random bit flip

Design Issues

This section will cover how the operations were used in building the heuristic method.

Reproduction through Fitness Values

The basic criterion for reproduction is the fitness value. Each string would be given a certain value depending on how "fit" they are, and the best values would be chosen to go to the mating pool to produce the next generation [1].

The first generation of bits is mostly random. Some strings in the first generation are obtained by methods for finding initial Basic Feasible Solutions. By doing this, it is guaranteed that the first generation will have some feasible solutions. The bits in a string corresponds to the variables in the objective function. Each string is given a fitness value depending on the feasibility of the solution. Depending on the size of the problem, a number of "best" strings are chosen to reproduce.

Crossover Operation

The subsequent generations are created by the method of crossover. In the GAs, there are two main kinds of crossovers:

1. **One-point crossover:** In this method, one point (the crossover point) is chosen in the parent strings and everything to the right of the crossover point is then interchanged to form the children nodes.

1101┃100 + 1000┃011=1101011+1000100

In this example, the bar is the crossover point, and the colored bits are interchanged in the children strings.

2. **Two-point crossover:** In this method, two crossover points are chosen in the parent strings. All the bits within the two crossover points remain unchanged. All the other bits are interchanged to produce the children.

   10┃011┃011+11┃101┃001=11011001+…

By placing the crossover points in different positions, many combinations of bits can be generated. Therefore, even from a small number of strings belonging to the first generation, a large number of children can be produced.

Mutation

The mutation operation is random flip of a bit. Since this operation is not very common, the heuristic generally has a mutation rate of 1% or less.

Solving Transportation Problems Using Genetic Algorithm

To solve Transportation Problems using this heuristic method, the following steps are needed:

1. Generating the Fitness Function

The fitness function is generated by multiplying the variables in the objective function by the corresponding supply amount (highest possible value for that variable). For example, the objective function for the problem in figure 2 is:

Min $z = 4X_{11} + 6X_{12} + 10X_{13} + 2X_{21} + 7X_{22} + 8X_{23}$.

To acquire the fitness function, we simply multiply the supply amount:

Min $z = 40X_{11} + 180X_{12} + 200X_{13} + 20X_{21} + 210X_{22} + 160X_{23}$

	Dest 1	Dest 2	Dest 3	Supply
Source 1	4	6	10	30
Source 2	2	7	8	30
Demand	10	30	20	

Figure 2

2. Creation of First Generation

The next step is to create the first generation of strings randomly. Then the strings with the lowest fitness values (since it is a minimization problem) are selected for reproduction.

3. Reproduction

The second generation is created by means of crossover and mutation operations. Then the best strings are chosen to create the next generation.

4. Check for Feasibility

The chosen strings are checked for feasibility of solution. To be a feasible solution, the string produced has to meet all the constraints. This is checked in this step. If a string is computationally infeasible, it is rejected.

5. Measure Value of Fitness Function

When a set of feasible strings is filtered from the entire generation, their corresponding objective values are measured. For example, the string 101010 is in essence,
$40(1) + 180(0) + 200(1) + 20(0) + 210(1) + 160(0) = 450$

This value (450) is then stored along with the string.

Steps 3 through 5 are then repeated for each subsequent generation of strings. After generating a sufficient number of generations, these objective values are compared and the lowest value found is declared as the optimal solution.

Findings

In comparison to the existing methods, Genetic Algorithms prove to be more efficient as the size of the problem becomes greater. For a 500×500 problem, the heuristic method is much faster than the Simplex and Transportation Simplex Methods. In terms of accuracy, the heuristic is 95.703% accurate.

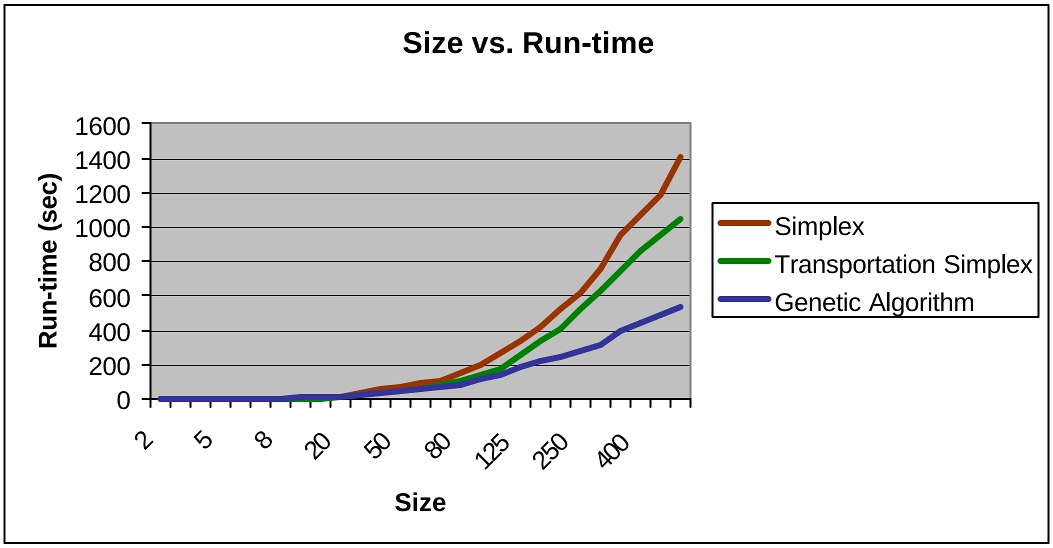

Figure 4

Advantages and Disadvantages

The GA-heuristic method is more efficient for large problems. Genetic Algorithm programs are simple to operate and have high reusability. Therefore, the same program can be used for other problems with minor changes [2].

One disadvantage of the heuristic method is the accuracy. As more generations are made, more accurate results are produced. So there is a trade off between run-time and accuracy.

Future research

In the design of the heuristic method, no parallelism was considered. With different generations being created simultaneously, the speed of the method can be greatly increased. As a part of continuing research, a parallel version of the algorithm will be designed.

References

1. Blum, C., Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR) 35*,2 (September 2003), 268-308.
2. Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company, Inc, 1989.
3. Hamdy, T. *Operations Research- An Introduction* (Sixth Edition). Pearson Education Inc, 2002.
4. Holland, J. H. *Genetic Algorithms*. (n.d.); see http://www.arch.columbia.edu/DDL/cad/A4513/S2001/r7/
5. Vose, Michael D. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, 1998.
6. Schilling, S. M., Nygard, K. E. Metastrategies for Heuristic Search. In *Proceedings of the Small College Computing Symposium* (1991), 213-222.
7. Winston, W. L. *Operations Research: Applications and Algorithms* (Third Edition). International Thompson Publishing, 1994.

Author Biography:

Tapojit Kumar (tapojit.kumar@mnsu.edu) is a senior year student in the Computer and Information Sciences department at Minnesota State University, Mankato, MN. His areas of interest include Combinatorial Optimization, Parallel Processing, Network Security and Algorithm Analysis.

Faculty Mentor Biography:

Susan Schilling (Susan.Schilling@mnsu.edu) has been a member of the faculty in the Department of Computer and Information Sciences at Minnesota State University since 1982. Her areas of interest include Operations Research, Computer Graphics, and introductory programming courses. She is currently studying the effectiveness of introducing robotics as a means of teaching programming to beginning Computer Science students.