Solving transportation problems with warehouse locations based on greedy algorithm

Xianyao Li

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 201800, China

524687207@qq.com

Keywords: Transportation problems, greedy algorithm, warehouses location.

Abstract. Greedy algorithm is a common-used method to solve many different types of problems. Researches were done a lot that use greedy algorithm to solve classical transportation problems. This paper discussed how to apply greedy algorithm in transportation problems with warehouses location. The main idea of the improved algorithm is adding average fixed costs to unit transport costs. Then we use this method to solve a realistic problem in a company called SportStuff.com, and we found that although greedy algorithm usually cannot give out the optimal solution, it could provide a satisfactory suboptimal solution simply and quickly. And when the scale of problem increases, its solution will become closer to the optimal one. However, it still has some other limitations in solving more realistic problems.

1. Introduction

1.1 Brief Introduction of Greedy Algorithm

Greedy algorithm is a simple and commonly used algorithm. Its main idea is to always make the best choice under the current state when we solve problems. That is to say, when solving each subproblems of the original problem, we always make a local optimal solution, until the end of the final problem. Because greedy algorithm does not consider a problem as a whole, the final solution often tends to be a local optimal solution to some extent.

Greedy algorithm for most problems can be summarized as the following main steps:

a) Express the original problem in the form of mathematical models, and make its goal and initial state clear;

b) Starting from the initial state, look for the the local optimal solution in this state;

c) Repeat until the end of the algorithm, finally get the solution.

Greedy algorithm is often used to solve the problems of some decisions in everyday life, such as change money problem (when a shop clerk wants to change money for customers, he always tends to pay the largest denomination coins firstly), knapsack problem (put several items with different weights and values into a knapsack whose loading capacity is limited, and maximize the total value of those items put into the knapsack), etc. Moreover, it is also applied in some more complex problems in other areas. Z Zhang used a greedy algorithm for aligning DNA sequences, and found it can be much faster than traditional dynamic programming approaches and yet produce an alignment that is guaranteed to be theoretically optimal[1]; R Ruizab applied a new iterated greedy algorithm to the permutation flowshop scheduling problem, which was both very simple to implement and highly effective[2]; C Chekuri, M Pal used a recursive greedy algorithm for walks in directed graphs to maximize the number of nodes visited[3].

1.2 Use Greedy Algorithm to Solve Transportation Problem

Transportation problem is a kind of common linear programming problems in operations research. In the problem, the assumption is that there are m supply points and n demand points, transportation costs of a unit product from each supply point to demand point are known, and the goal is to find the optimal transport plan that minimize the total cost under the restrictions of supply and demand. Usually, we can use some linear programming algorithm to solve it, but we can also use some simple heuristic algorithms such as greedy algorithm to solve such problems.

2. Method

2.1 Steps to Solve Transportation Problem

a) Draw up a table of the transportation model, the transverse represents supply sources (factory), and longitudinal axis represents demand sources (warehouse). Then, fill the upper left corner in each grid with the unit transportation cost from corresponding supply point to demand point, while the number in the center of each grid represents the flow condition from each supply point to demand point.

b) According to the greedy algorithm, find the smallest unit transportation cost in the table.

c) In the grid with the smallest unit cost, write down the minimum number between its corresponding quantities of supply and demand.

d) Draw a line on the saturated rows or columns, means that all the products in the supply point have been already sent out or all the products the demand point needs have come in. And those rows or columns with lines are never taken into consideration after that.

e) Repeat step b) ~ d) until it meets all the constraints, then we get the final solution.

In supply chain planning, sometimes the designer will consider to construct some new supply chains. In that case, they will also need to consider the selection of warehouses (supply points) sites and the new costs brought by constructions of the new warehouses. But in the use of greedy algorithm to solve transportation problem, we assume that all warehouses have been built and can be used, so only transportation costs are under consideration, thus ignoring the variable costs and fixed costs.

To make the greedy algorithm continue to work, the concentration is focused on how to convert the fixed costs to transport costs in a reasonable way. Suppose each warehouse holds average storage inventory (generally half of the maximum inventory), we can calculate the average fixed cost which per unit of inventory holds, then add it to the the original transport costs, thus forming a new transportation cost table. Finally, using the new cost table, we can still use the conventional greedy algorithm to solve this kind of transportation problems.

2.2 Application Example

We use the chain management project of Spotstuff.com as an example. Sportstuff.com is a company who recycles and sales second-hand children's sporting equipment, the company expects that in the next three years (2000-2002), the sales will increase by 80% each year. And the original stock condition cannot meet the needs of the growing, so we must to redesign the company's supply chains. The goal is to minimize the total cost of the supply chain, including warehouse holding costs, fixed costs, transportation costs and variable costs.

The following is data we have known already:

a) Regional Demand at SportStuff.com for 1999:

Table 1. Regional Demand in1999

Zone	Demand in1999	Zone	Demand in 1999
Northwest	320,000	Lower Midwest	220,000
Southwest	200,000	Northeast	350,000
Upper Midwest	160,000	Southeast	175,000

b) Transportation costs are showed in table 2. To offset the cost, the company will charge 3 dollars to customers before every shipment.

Table 2. Transportation costs per shipment (Four Units)

	Northwest	Southwest	Upper Midwest	Lower Midwest	Northeast	Southeast
Seattle	2.00	2.50	3.50	4.00	5.00	5.50
Denver	2.50	2.50	2.50	3.00	4.00	4.50
St.Louis	3.50	3.50	2.50	2.50	3.00	3.50
Atlanta	4.00	4.00	3.00	2.50	3.00	2.50
Philadelphia	4.50	5.00	3.00	3.50	2.50	4.00

c) Fixed and Variable Costs of Potential Warehouses:

Table 3. Fixed and Variable Costs

	Small Warehouse		Large Warehouse	
	Fixed Cost ($/year)	Variable Cost ($/Unit Flow)	Fixed Cost ($/year)	Variable Cost ($/Unit Flow)
Seattle	300,000	0.2	500,000	0.2
Denver	250,000	0.2	420,000	0.2
St.Louis	220,000	0.2	375,000	0.2
Atlanta	220,000	0.2	375,000	0.2
Philadelphia	240,000	0.2	400,000	0.2

Small warehouses could handle a flow of up to 2 million units per year, whereas large warehouses could handle a flow of up to 4 million units per year.

d) Holding Costs

The inventory holding costs at a warehouse (excluding warehouse expense) was about $\$600\sqrt{F}$, where F is the number of units flowing through the warehouse per year.

3. Result

a) First of all, share the fixed costs equally into the transportation costs

Table 4. Average fixed costs

	Small Warehouse		Large Warehouse	
	Fixed Costs	Average Fixed Costs	Fixed Costs	Average Fixed Costs
Seattle	300,000	1.2	500,000	1
Denver	250,000	0.5	420,000	0.84
St.Louis	220,000	0.88	375,000	0.75
Atlanta	220,000	0.88	375,000	0.75
Philadelphia	240,000	0.96	400,000	0.8

For example, $1.2 = \frac{300,000}{(2,000,000 \div 2) \times 4}$. By comparing the costs, we can presuppose the kind of warehouse we will rent or build, and according to the final usage we can change big warehouse to small warehouse. The presupposes in each location are showed in table 5.

Table 5. Presupposes of Warehouse

Seattle	Denver	St.Louis	Atlanta	Philadelphia
Large Warehouse	Small Warehouse	Large Warehouse	Large Warehouse	Large Warehouse

b) Add the average fixed costs to transportation costs, we can get the new transportation cost table.

Table 6. New Transportation Cost Table

	Northwest	Southwest	Upper Midwest	Lower Midwest	Northeast	Southeast
Seattle	3.00	3.50	4.50	5.00	6.00	6.50
Denver	3.00	3.00	3.00	3.50	4.50	5.00
St.Louis	4.20	4.25	3.25	3.25	3.75	4.25
Atlanta	4.75	4.75	3.75	3.25	3.75	3.25
Philadelphia	5.3	5.80	3.80	4.3	3.3	4.8

c) Use greedy algorithm to find out the solutions in the next 3 years.

Table 7. Transport Planning in 2000

	Northwest	Southwest	Upper Midwest	Lower Midwest	Northeast	Southeast
Seattle						
Denver	576,000	360,000	288,000	396,000		
St.Louis						
Atlanta						315,000
Philadelphia					630,000	

The solution is to rent small warehouse in Denver, Atlanta and Philadelphia, and the total cost is $2528535.

Table 8. Transport Planning in 2001

	Northwest	Southwest	Upper Midwest	Lower Midwest	Northeast	Southeast
Seattle						
Denver	1,036,800	648,000	315,200			
St.Louis			203,200	712,800		
Atlanta						567,000
Philadelphia					1,134,000	

The solution is to rent small warehouse in Denver, St.Louis, Atlanta and Philadelphia, and the total cost is $3967984.

Table 9. Transport Planning in 2002

	Northwest	Southwest	Upper Midwest	Lower Midwest	Northeast	Southeast
Seattle		1,032,760				
Denver	1,866,240	133,760				
St.Louis			933,120		344,840	
Atlanta				1,283,040	1,696,360	1,020,600
Philadelphia						

The solution is to rent small warehouse in Seattle, Denver, St.Louis, rent large warehouse in Atlanta, and the total cost is $5360012.

4. Conclusion

In order to judge the solution of the greedy algorithm, we also use the linear approximation, the genetic algorithm and particle swarm optimization (PSO) algorithm, using matlab programming, to respectively search for the optimal solution of this case. Table 10 shows the total cost of the solutions of different methods.

Table 10. Total Cost of each Solution

	Greedy Algorithm	Linear Approximation	Genetic Algorithm	Particle Swarm	Deviation between the Optimal one
2000	2,528,535	2,018,812	3,849,083	2,569,431	25.24%
2001	3,967,984	3,420,507	5,349,004	3,384,681	17.23%
2002	5,360,012	5,038,751	7,713,573	4,640,899	15.49%

It indicates that the greedy algorithm is difficult to ensure the final solution being the best one. In another word, using other heuristic algorithms could often find a better solution to the problem. Firstly, this is because greed algorithm usually chooses the local optimal choice, instead considering it as a whole. Secondly, in the operation process of the greedy algorithm, due to the limitation of algorithm itself and ignorance of holding costs and variable costs, it will have some impact on the results to some extent. But it does not mean that those algorithms which didn't give out a fine solution is useless, after some adjustments and improvements, they could also be efficient and effective. Lideng Pan presented a simple heuristic greedy method to solve TSP, in which only distances between all the cities are used [4]. Youzhao Wang combined greedy algorithm and genetic algorithm to schedule vehicles in warehouses and experimental analysis demonstrated that it could promote the efficiency of scheduling [5].

However, the deviation between the result of the greedy algorithm and the optimal solution is acceptable (compared to all other heuristic algorithm solutions), it is because the greedy algorithm focuses on transportation costs and fixed costs, especially transportation costs, which is the key point to influence the total cost. If we could control transportation costs and fixed costs to a relevant lower level, even the final solution is a local optimal solution, it should be an acceptable solution. Table 9 also shows that, as the demand increases year by year, gradually increasing the scale of the problem, the deviation between the greedy algorithm solution and the optimal solution is also gradually narrowed. It seems that greedy algorithm may have advantages in large-scale transportation problems. In a larger scale, design and execution of other heuristic algorithms will become more complicated and time-consuming, while the greedy algorithm appears simpler and quicker, and can provide a satisfying suboptimal solution.

But using the greedy algorithm to solve the transportation problems with site selection also has some weaknesses. For instance, it can only handle the situation that only build one warehouse at one place. But in realistic problems, constructing multiple warehouses is possible and common. However, the greedy algorithm in this paper is unable to solve those problems with more than one warehouses in a same place.

References

[1]. Zhang Z, Schwartz S, Wagner L, et al. A greedy algorithm for aligning DNA sequences [J]. Journal of Computational Biology a Journal of Computational Molecular Cell Biology, 2000, 7(1-2):203.

[2]. Ruizab R. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem [J]. European Journal of Operational Research, 2007, 44(3):2033-2049.

[3]. Chekuri C, Pal M. A recursive greedy algorithm for walks in directed graphs[C]// Foundations of Computer Science, 2005. FOCS 2005. IEEE Symposium on. DBLP, 2005:245-

[4]. Lideng Pan, Xiaofeng Huang. Solving TSP with a simple heuristic greedy method [J]. Journal of Beijing University of Chemical Technology, 1998(2):46-51.

[5]. Youzhao Wang. Scheduling algorithm for vehicles in warehouses based on greedy algorithm and genetic algorithm [J]. Transducer and Microsystem Technologies, 2012, 31(10):125-128.