2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.40

Model and Algorithm for Computer Spare Parts Logistics Management System

Huihui Huang+ and Buyang Cao

School of Software Engineering, Tongji University, Shanghai, China

Abstract. How to provide customers with timely and effective logistics and distribution services as well as reduce the production cost of the sales process has become increasingly important for enterprises in recent years. This paper is a study on computer spare parts logistics management information system project. It describes the project background and the composition of the system. Research is carried out on how to handle the distribution model with multiple types of nodes and more complex spare parts logistics. The open-source Mixed Integer Linear Programming (MILP) solver—lp_solve is used to solve the model, and the global optimal solution can be reached by employing branch-and-bound methodology. Finally this paper proposes a heuristic algorithm to meet the business needs and quickly get the approximate result.

Keywords: Transportation problem; integer programming; greedy algorithm; branch-and-bound algorithm; logistics system; distribution model; lp_solve

1. Introduction

Spare parts logistics is a comprehensive logistics activity which is aiming to ensure timely and efficient supply of spare parts, and to provide enterprises with support to normal operation and after-sales service.

With more competitors in the market, customers are able to choose more and more similar products. It is quite often that individual products have little difference, especially in personal PC. Therefore after-sale service becomes an important business tool for competitive differentiation [1][2]; and after-sale service is a major factor for customer purchase decision. On the one hand the success of spare parts logistics can improve customer satisfaction and loyalty and thereby establish a good corporate reputation and indirectly increase the profits of enterprises; on the other hand, the success of the spare parts logistics management system can effectively control costs, reduce the costs and risks due to product backlog and/or overstock. Furthermore, the information in the system can help optimizing transportation routes, reducing transportation costs, directly improving the operational efficiency of enterprises. As a support service for spare parts logistics, operational efficiency of its treatment often determines the service quality, speed and cost, and therefore become very important [3].

This paper is written based on an IT spare parts logistics management system project. The main goals of the paper focus on establishing the model and developing corresponding algorithm to find reasonable distribution routes. There are plenty of related researches, but few of them were integrated into a completed logistics management system as a distribution optimizer. This optimizer enhances the intelligence of the overall logistics management, and makes the whole distribution process more optimized [4].

+ Corresponding author.
E-mail address: hhhaui@163.com.

2. Background

2.1 System Architecture

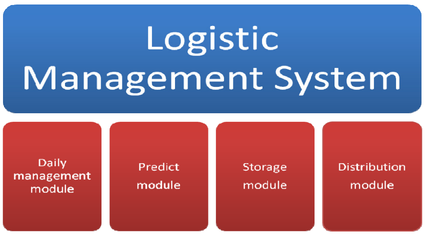

Figure 1. System Architecture

The system consists of the following major components:

- Daily Management Module: The module is responsible for managing the daily plans and tasks;
- Predict Module: it forecasts the purchasing quantities and the intervals of purchasing according to the historical procurement data;
- Storage Module: The module is responsible for managing the storing data and checking-in/-out registration;
- Distribution Module: The module queries inventory and request information according to a number of required inputs, and then builds the optimal delivery routes for the delivery schedulers;

2.2 Distribution module

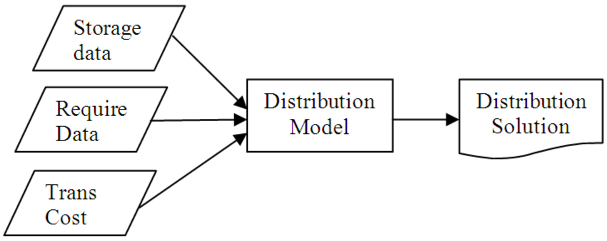

Figure 2. Flow chart of distribution module

The distribution module will perform the following steps in order to generate a delivery plan:

- Store data: The data is extracted from the database of the inventory management module;
- Require data: The data is generated automatically by the management system based on the submitted reports, or directly acquired from users' inputs.
- Compute cost Matrix: The matrix represents a cost rate matrix between any pair of locations. It can be maintained and updated by a user based upon the current tariff in order to perform more accurate calculation.

3. Mathematical Model

3.1 Transportation problem

Transportation problem: the transportation problem is a classical minimum-cost flow problem. In linear-programming terms, we are given a m-entry supply vector A={a[i]}, an n-entry demand vector B={b[j]}, and an $m \times n$ cost array C={c[i, j]}, such that the entries of A and B are all positive, and we want to choose an $m \times n$ variable array X={x[i,j]} in order to

$$\text{MIN } Z = \sum_{i,j} c(i,j) \times x(i,j)$$

$$\textit{subject to}: \sum_{j} x[I,j] \le a[I] \qquad \textit{for } 1 \le I \le m$$

$$\sum_{i} y[i,J] \ge b[J] \qquad \textit{for } 1 \le J \le n$$

3.2 Transportation problem include transit node

We can set various constraints according to the types of transit nodes [5]:

- Pure Supply Node: A pure supply node with supply s[I] doesn't have demand nor perform transfer function;

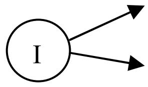

Figure 3-1. Pure Supply Node

Constraints: The total amounts sent to other nodes from a pure supply node I must less than the supply amount s [I]

$$\sum_j x[I, j] \le s[I]$$

- Pure Demand Node: a pure demand node with the demand quantity of d[I] will not have any supply and won't perform transfer function;

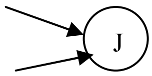

Figure 3-2. Pure Require Node

Constraints: The total amount received from other nodes must meet the J node's requirement

$$\sum_i x[i, J] \ge d[J]$$

- Supply Node: a supply node with supply quantity of s[I] doesn't have any demand but it might perform transfer function;

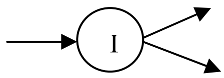

Figure 3-3. Supply Node

Constraints: outflow - inflow must be less than supply amount possessed by node I:

$$\sum_j x[I, j] - \sum_i x[i, I] \le s[I]$$

- Demand Node: a demand node with demand quantity of d[I] will not provide any supply but it might perform transfer function;

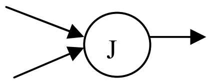

Figure 3-4. Require Node

Constraints: inflow - outflow must be greater than the demand quantity required by node J

$$\sum_i x[i, J] - \sum_j x[J, j] \ge d[J]$$

- Pure Transit Node: this type of nodes only perform transfer function, and they don't have demand nor supply;

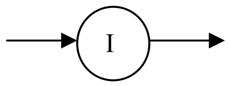

Figure 3-5. Pure Transit Node

Constraints: there is no capability of storage, inflow is equal to outflow.

$$\sum_i x[i, I] - \sum_j x[I, j] = 0$$

- Transit Node: this type of nodes may have demand, supply and perform transfer function;

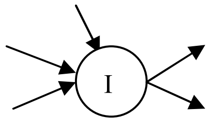

Figure 3-6. Transit Node

Constraints: inflow - outflow must not be greater than the amount of supply – demand;

$$\sum_i x[i,I] - \sum_j x[I,j] \le s[I] - d[I]$$

3.3 Notation

- $x^k_{i,j}$: The amounts of the k-th spare part to be transported from location *i* to location *j*;
- $c^t_{i,j}$: The matrix of unit transportation cost from location *i* to location *j* with the t-th transportation method;
- H: The set of all of the departure ports (including all nodes with capability of storage);
- D: The set of all of the destinations;
- M: The set of the pure transit node;
- K: The set of the spare parts;
- $d_{j,k}$: The require of location j for k-th spare parts;
- $s_{i,k}$: The current storage of k-th spare parts at location i;
- $\min s_{i,k}$: The minimum storage amount of k-th spare parts at location i.

3.4 Distrubution model

In this problem, departure ports include main-HUBs and sub-HUBs ,and destination only include sub-HUBs.all the main-HUBs can be treated as the supply nodes, sub-HUBs can be viewed as the transit nodes with capacity, other nodes are pure cargo transit nodes without storage capacity [6][7].

The objective Function is to minimize the total transporting cost, i.e.:

$$\text{MIN } Z = \sum_{i\in H\cup M} \sum_{j\in D\cup M} \sum_{k\in K} c^k_{i,j} x^k_{i,j}$$

Subject to:

1) *Require constraints:*

$$\sum_{i\in H} x^k_{i,j} \ge d_{j,k}, \forall J \in D, \ \forall k \in K$$

2) *Supply constains(main-HUBs+sub-HUBs):*

$$\sum_{j\in D} x^k_{i,j} \le s_{i,k} - \min s_{i,k}, \forall J \in D, \ \forall k \in K$$

3) *Transit node constraints (sub-HUBs):*

$$\sum_{i\in H} x_{i,J} - \sum_{j\in D} x_{J,j} \ge s_{J,k} - d_{J,k} - \min s_{J,k}$$

$$\forall m \in M, \forall J \in D, \ \forall k \in K$$

4) *Pure transit node constraints:*

$$\sum_{i\in H} x^{k\partial}{}_{i,m} - \sum_{j\in D} x^{k}{}_{m,j} = 0 \quad \forall m \in M, \ \forall k \in K$$

5) *Non-negative and integer constraints:*

$$x^k_{i,j} \ge 0$$

4. Implementation

4.1 Implementation of the algorithm

As the nature of IT parts, the amounts of spare parts cannot be decimal, so the transpotation problem can be considered as an integer programming problem. Furthermore, branch-and-bound method which is widely used to solve integer programming problem, is also applicable for solving the mathematical model listed above.

In the actual implementation process, there are 13 HUBs, 2000 kinds of spare parts and more than 300 transit points. The number of variable $x_{i,j}^k$ is:

$$(13+300)*(12+300)*2000= 195312000$$

The optimal problem (of millions of integer variables) could be solved by a tranditional branch and bound method. We simply invoke the interger programming model in an open source package called “lp_solve” to get the optimal solution. The computing time on a Windows server is about 10 minutes which has been accepted by our customers with great satisfaction!

4.2 Approximate solution uinsing greedy algorithm

Considering the performance, the approximation algorithm or “greedy algorithm” could be used to solve even larger problems as the business grows and the number of hubs and sub-hubs increases rapidly.

Although the global optimal solution cannot be obtained by the greedy algorithm, the approximate optimal solution is valuable to the most practical projects that require for simple, fast, and reasonable solutions.

The greedy algorithm embedded in “lp_solve” is applied to solve the same problem, and the computational time is reduced to 2 minutes.

Due to more locations being added to the existing distribution network, the traditional branch-and-bound approach is unable to solve the corresponding problems within reasonable computational time. The greedy algorithm will play an important role in the practice to provide acceptable and feasible solutions to the practitioners.

5. Acknowledgment

Of the research is partially sponsored by the Key Project of the National Eleventh-Five Year Research Program of China (2006BAJ11B). And we’d like to thank all students in the project team for the joint efforts: SHEN Yong Yuan, FAN Xingang, FAN Qimeng, SHI Yuanchao, and LI Jinghui.

This paper is sponsored by Key Project of the National Eleventh-Five Year Research Program of China (2006BAJ11B).

6. References

[1] Wan Fengjiao, Zhang Qingnian. Study on the location routing problem of the multi-stage logistics network. IEEE, 2008 3-21.

[2] Hillegersberg, J. v., R. Zuidwijk, et al. (2001). "Supporting return flows in the supply chain." Commun. ACM 44(6): 74-79.

[3] Laporte G., Nobert Y. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research, 1981, Vol.6, pp224-236.

[4] Eilon,S. ,Watson-Gandy, C.D.T., Christofides, N., Distribution Management, Mathematical Modeling and Practical Analysis, Griffin, London, 1971

[5] Leon Cooper. The transportation-location problem. Operations Research, 1972, Vol.20, pp94-108.

[6] Lee, J.-e., M. Gen, et al. (2008). Designing a multistage reverse logistics network problem by hybrid genetic algorithm. Proceedings of the 10th annual conference on Genetic and evolutionary computation. Atlanta, GA, USA, ACM: 1707-1708.

[7] Mula, J., D. Peidro, et al. (2010). "Mathematical programming models for supply chain production and transport planning." European Journal of Operational Research 204(3): 377-390.