

A Very Large Scale Neighborhood (Vlsn) Search Algorithm for an Inventory-Routing Problem

Sombat Sindhuchao

Department of Industrial Engineering,
Ubonratchathani University, 34000
somsin@rocketmail.com

Abstract

An inventory-routing problem is one of the combinatorial optimization problems that are very difficult to solve optimally. The inventory-routing system considered is composed of a central warehouse and a set of suppliers who produce non-identical items. The items are kept at the central warehouse and face constant and deterministic demands. An inventory policy with fixed order quantity is adopted. When a group of items need to be replenished, a capacitated vehicle is dispatched from the central warehouse to collect them at suppliers' locations. In this paper, constructive heuristics and an improvement algorithm called a very large scale neighborhood (VLSN) search algorithm are presented. The constructive heuristics sequentially generate an initial feasible solution. Then, a VLSN search algorithm is applied to improve the solution by searching very large neighborhoods. Computational results indicate that the VLSN search algorithm can provide near-optimal solutions very efficiently.

Keywords: Inventory-routing; a very large scale neighborhood search algorithm; fixed order quantity.

1. Introduction

This paper focuses on an inbound commodity collection system, where items are collected from suppliers and stored at a warehouse. The system considered is composed of a central warehouse and a set of suppliers who produce non-identical items. The items are kept at the central warehouse and face constant and deterministic demands from outside retailers. The items are jointly replenished using economic order quantity (EOQ) inventory policy. When a group of items need replenishment, a capacitated vehicle is dispatched from the central warehouse to collect them at suppliers' locations. After completing item collection, the vehicle returns to the warehouse. The vehicle can be dispatched at any time but it cannot travel more than a maximum number of trips allowed per year.

The decision is to determine the replenishment quantity and replenishment interval for each item, along with an efficient vehicle route so as to minimize the total inventory and transportation costs per unit time.

This is an inventory-routing problem which is one of combinatorial optimization problems. This problem is NP-hard. Therefore, it is very difficult to obtain the optimal solution. However, heuristics can be applied to find a near-optimal solution with reasonable time.

In this paper, constructive heuristics and an improvement algorithm called a very large scale neighborhood (VLSN) search algorithm are presented. The constructive heuristics sequentially generate an initial feasible solution. Then, a VLSN search algorithm is applied to improve the solution by searching very large neighborhoods.

A number of related papers considering the integration of inventory and routing are available. Most of them involve distribution of a single item. Federgruen and Zipkin [10] are the first to integrate the allocation and routing problems in a single model. They consider allocation of a scarce resource from a central depot to many retailers using a fleet of capacitated vehicles. Random demands in a single period model are assumed. They

formulate the problem as a non-linear integer program and interchange heuristics are modified to solve the deterministic vehicle routing problem. A single-item distribution system with one depot and a set of retailers that keep inventories is considered by Anily and Federgruen [5]. They assume that the demand rate of each retailer is an integer multiple of some base rate. Anily [6] extends their work by employing general holding cost rates. Bertazzi, Paletta and Speranza [7] propose a heuristic algorithm for solving an inventory-routing problem in which a deterministic order-up-to level policy is adopted and Cousineau-Ouimet [8] develops a Tabu Search algorithm to resolve the adapted instances of Bertazzi, Paletta and Speranza [7]. The results from both algorithms are relatively similar. For a multi-item case, Viswanathan and Mathur [15] integrate a vehicle routing problem and inventory decisions in a single warehouse multi-retailer multi-item distribution system with deterministic demands. They propose a stationary nested joint replenishment policy (SNJRP) heuristic to solve the problem where replenishment intervals are limited to be power of two multiples of a base planning period. A heuristic decomposition method to solve an integrated inventory and transportation problem is presented by Qu, Bookbinder and Iyogun [12]. They adopt a modified periodic policy in which each joint replenishment period is an integer multiple of a base period.

The remainder of this paper is organized as follows. In section 2, the problem is described and the model is formulated. In section 3, constructive heuristics and a very large scale neighborhood (VLSN) search algorithm are presented. Computational tests are reported in section 4. Finally, this research is concluded in section 5.

2. Problem Description and Formulation

An inbound commodity collection system in a deterministic setting is considered. The system consists of a central warehouse and a set of suppliers each of whom produces one or more non-identical items. Each item faces constant and deterministic demand from outside retailers. The central warehouse has unlimited space for stocking items that are jointly replenished using economic order quantity (EOQ) inventory policy. A policy where the set of items is

partitioned into disjoint groups is adopted. Each group of items is assigned to a vehicle. When each group of items needs replenishment, a vehicle is dispatched to collect them from their suppliers. It is assumed that there are an unlimited number of identical vehicles available at the central warehouse. These vehicles have limited capacity and cannot travel more than a maximum number of trips allowed per year. The costs of the system include inventory holding costs at the central warehouse, fixed joint ordering costs, fixed dispatching costs and vehicle routing costs.

Notation

To formulate the mathematical model, the following notation is defined.

Constant and Indices

S	set of items in the system.
m	number of available vehicles.
n	total number of items.
i	index for vehicles ($i = 1, 2, \dots, m$).
k	index for vehicles ($k = 1, 2, \dots, m$).
j	subscript denoting item ($j = 1, 2, \dots, n$).
D_j	demand rate for item j .
h_j	inventory holding cost rate for item j .
K	fixed joint ordering cost plus fixed dispatching cost.
C	vehicle capacity.
F	maximum number of trips allowed for each vehicle.

Variables

S	subset of items $S \subseteq S$.
$TSP(S)$	optimal vehicle route for visiting suppliers of items in subset S .
$L(S)$	fixed transportation costs plus fixed joint ordering cost of items in subset S .
$D(S)$	aggregate demand rate for all items in subset S .
$h(S)$	weighted average unit holding cost for items in subset S .
$T(S)$	replenishment interval for items in subset S .
$Q(S)$	aggregate replenishment quantity for all items in subset S .

- Q_j replenishment quantity of item j .
- Q_j^* optimal replenishment quantity of item j .
- $Q^*(S)$ optimal aggregate replenishment quantity for all items in subset S .
- $T^*(S)$ optimal replenishment interval for all items in subset S .
- $S^{(i)}$ subset of items assigned to vehicle i .
- $c(S)$ average total inventory-transportation cost for all items in subset S .

To formulate the mathematical model of this problem, a single subset S of items is considered. Therefore, the average total inventory-transportation costs per unit time for items in subset S can be written as:

$$c(S) = L(S) \frac{D(S)}{Q(S)} + \frac{1}{2} h(S) Q(S) \quad (1)$$

where $L(S) = K + TSP(S)$ (2)

$$D(S) = \sum_{j \in S} D_j \quad (3)$$

$$Q(S) = \sum_{j \in S} Q_j \quad (4)$$

$$h(S) = \frac{\sum_{j \in S} h_j D_j}{D(S)} \quad (5)$$

$$T(S) = \frac{Q(S)}{D(S)} \quad (6)$$

Due to the vehicle capacity and frequency constraints, the aggregate demand rate for all items in subset S must not be greater than CF to ensure feasibility. The objective is to determine the replenishment quantity and replenishment interval for each item along with an efficient vehicle route so as to minimize the total inventory and transportation costs per unit time. The optimal aggregate replenishment quantity $Q^*(S)$ for items in subset S can be calculated from:

$$Q^*(S) = \begin{cases} \frac{D(S)}{F} & \text{if } \sqrt{\frac{2D(S)L(S)}{h(S)}} \leq \frac{D(S)}{F} \\ \sqrt{\frac{2D(S)L(S)}{h(S)}} & \text{if } \frac{D(S)}{F} \leq \sqrt{\frac{2D(S)L(S)}{h(S)}} \leq C \\ C & \text{if } C \leq \sqrt{\frac{2D(S)L(S)}{h(S)}} \end{cases} \quad (7)$$

Once the optimal aggregate replenishment quantity is known, the optimal replenishment interval $T^*(S)$ and the optimal replenishment quantity Q_j^* of item j in this subset can be obtained from:

$$T^*(S) = \frac{Q^*(S)}{D(S)} \quad (8)$$

$$Q_j^* = D_j T^*(S) \quad (9)$$

For a set S of all items in the system, the problem can be formulated as a partitioning problem. Assuming that there are m vehicles, the problem becomes:

$$\min \sum_{i=1}^m c(S^{(i)}) \quad (10)$$

subject to

$$\bigcup_{i=1}^m S^{(i)} = S \quad (11)$$

$$S^{(i)} \cap S^{(k)} = \phi \text{ for all } i, k = 1, 2, \dots, m; i \neq k. \quad (12)$$

$$D(S^{(i)}) \leq CF \text{ for all } i = 1, 2, \dots, m \quad (13)$$

3. Solution Methods

The inventory-routing problem is a combinatorial optimization problem. As a result, it is extremely hard to find the optimal solution. However, heuristics can be applied to find a near-optimal solution within reasonable time. In this section, constructive heuristics and a very large scale neighborhood (VLSN) search algorithm are presented. Constructive heuristics are performed to partition items into groups. A vehicle route for each subset can be initially designed by applying the arbitrary insertion heuristic (see Rosenkrantz et al. [13]) and the TSP tour is improved by using the 2-opt exchange heuristic (see Croes [9] and Lin [11]). The inventory-routing problem can now be solved for each subset to obtain an initial solution. Then, the VLSN algorithm is applied to improve the solution.

3.1 Replenishment Interval (RI) Based Heuristic

In the RI heuristic, a vehicle route is sequentially constructed for one vehicle at a time. The replenishment interval of each individual item is used as the selection value for assigning an item to the vehicle. The assignment of items to the vehicle is based on an idea that items with similar replenishment interval should be replenished at the same time. The procedure of this heuristic is described below.

- Step1. Solve for a replenishment interval T_i of every item i individually and list the items in the increasing order of the replenishment interval.
- Step2. Initialize an empty route for the next vehicle.
- Step3. For ungrouped items, assign one item at a time to the vehicle, starting from the first ungrouped item in the list (the one with smallest replenishment interval). Continue adding an ungrouped item to the vehicle until it cannot be added due to the capacity and frequency constraints.
- Step4. If all items have been assigned to a vehicle, go to step 5. Otherwise, return to step 2.
- Step5. Find a TSP tour for all vehicles.

3.2 Distance Sum (DS) Heuristic

The DS heuristic also constructs routes sequentially for one vehicle at a time. In this heuristic, transportation costs are implicitly considered for item selection. The idea of item assignment is that the best candidate should be an item whose supplier is located closest to the warehouse and also close to at least one supplier that is already in the route. Let $d_{i/2}$ denote the distance (cost) from the supplier of item j_1 to the supplier of item j_2 , for all $j_1, j_2 \in S$.

Similarly, let d_{0j} and d_{j0} denote the distance from the warehouse to the supplier of item j .

- Step1. Initialize an empty route for the next vehicle.
- Step2. For each ungrouped item j that can be added to the vehicle without violating the capacity constraint, determine its

distance-sum as the minimum value of $d_{ji} + d_{0j}$ over all items j served by the current vehicle. If the current vehicle does not contain any items, let the distance-sum be d_{0j} . If no such items exist, go to Step 4.

- Step3. Find the item with the smallest distance-sum, assign it to the vehicle, and return to Step 2.
- Step4. If all items have already been assigned to vehicles, go to Step 5. Otherwise, if there are available vehicles left, return to Step 1.
- Step5. Find a TSP tour for all vehicles.

3.3 Very Large Scale Neighborhood (VLSN) Search Algorithm

For neighborhood search algorithms, if the neighborhood is very large, a very good and accurate final solution can be obtained. However, at each iteration, it will take a very long time to search for the best solution in the neighborhood. Very large-scale neighborhood search algorithms have very large neighborhoods, but an improved neighbor can be identified quickly without explicit enumeration and evaluation of all neighbors in the neighborhood (see Ahuja, Orlin and Sharma [4]). This efficient method for identifying an improved neighbor is based on a characterization of the neighborhood through an improvement graph (see Ahuja, Ergun, Orlin and Punnen [1]). Like any other neighborhood search algorithms, VLSN starts with an initial feasible solution and keeps searching for a better solution until no improvement is obtained. In this problem, items are considered as nodes in constructing an improvement graph. The neighborhood of the solution can be reached by moving a single item from one vehicle to another vehicle. The number of vehicles involved in the movement of items can be more than two. This means that a group of items, each of which belongs to different vehicles, can move simultaneously from its current vehicle to another vehicle. The type of exchange is called a cyclic exchange if an item moves from the last vehicle to the first vehicle involved in the item movement. It is called a path exchange if no item moves out from the last vehicle and no item moves into the first vehicle. For application of VLSN, see Ahuja and Orlin [3] and Ahuja, Jha,

Orlin and Sharma [2]. The general procedure can be described below.

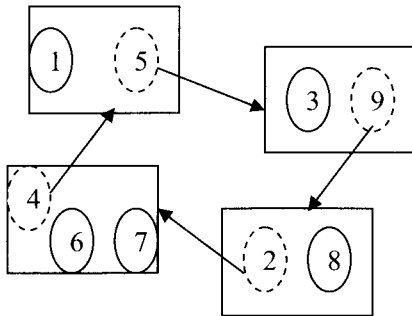


Figure 1. Cyclic Exchange

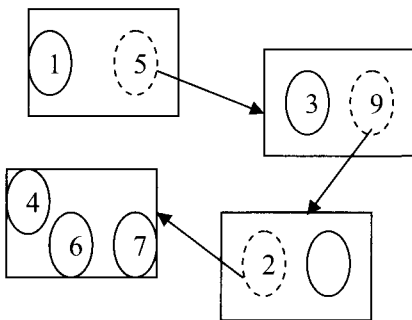


Figure 2. Path Exchange

- Step 1. Generate an initial feasible solution using one of proposed heuristics.
- Step 2. Construct the improvement graph for the cyclic exchange and path exchange neighborhoods.
- Step 3. Identify a cost-decreasing cyclic exchange or path exchange. If one of them exists, improve the solution by moving items according to the cost-decreasing exchange identified and return to step 2. Otherwise, stop.

4. Computational Tests

In order to compare the performance of the proposed heuristics, computational tests are performed on randomly generated problem sets. The demand rate and the inventory holding cost rate for each item are randomly generated from

the uniform distribution on $[100,300]$ and $[1,15]$ respectively. The items are randomly assigned to 10 suppliers in such a way that each supplier must produce at least one item. The locations of the warehouse and suppliers are generated uniformly in the square $[0,20]^2 \subset \mathbb{R}^2$, and Euclidean distances are used to measure vehicle routing costs, with unit cost per unit distance traveled. For every instance, the vehicle capacity C is set to 150 units with $F = 10$ trips per time unit. The fixed ordering and fixed dispatching costs are combined as $K = 50$. The size of an instance is identified by the number of items, n , and the number of vehicles, m . Six different problem sizes ($(n=15, m=3)$, $(n=30, m=6)$, $(n=40, m=8)$, $(n=50, m=10)$, $(n=100, m=20)$ and $(n=200, m=40)$) are tested with ten instances for each problem size. The heuristics have been coded in the C++ programming language on a PC with a 866 MHz Intel Pentium 3 CPU and 256 MB of RAM.

The solutions from the first four problem sizes are compared to the column generation lower bound on the optimal costs (see Sindhuchao [14]). Figures 3-6 display the percent error of the objective function values from RI, DS, RI-VLSN and DS-VLSN with respect to the lower bound. For the large scale problems ($n=100, m=20$) and ($n=200, m=40$), the objective function values obtained from RI, DS, RI-VLSN and DS-VLSN are compared to the best values among them. The results are given in Tables 1-2. The computational time of every heuristic is shown in Table 3.

From the computational results, the DS heuristic outperforms the RI heuristic. In other words, using transportation costs for partitioning items into subsets provides better initial solutions than considering replenishment intervals. As a result, the DS-VLSN also outperforms the RI-VLSN in most cases. The gap between the solutions obtained from RI and DS becomes larger as the problem size increases. For all instances, the error bounds of the solutions obtained from the DS-VLSN are about 3% on average. In addition, both RI-VLSN and DS-VLSN can find a solution very rapidly. For the large scale problems, ($n=100, m=20$), and ($n=200, m=40$), the DS-VLSN still outperforms the RI-VLSN in all cases. Moreover, the computational time of the DS-VLSN is about a half of the RI-VLSN's.

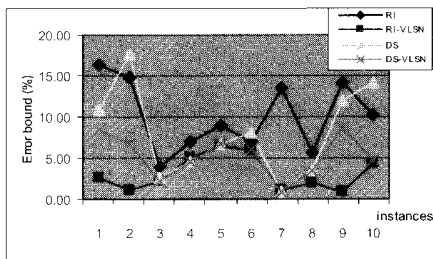


Figure 3. Error bound of solutions from RI, DS, RI-VLSN and DS-VLSN (n=15, m=3)

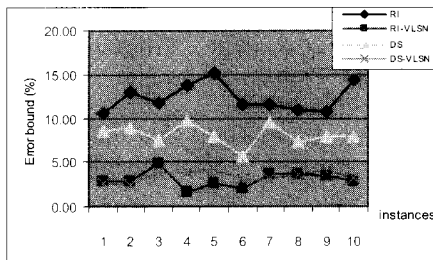


Figure 4. Error bound of solutions from RI, DS, RI-VLSN and DS-VLSN (n=30, m=6)

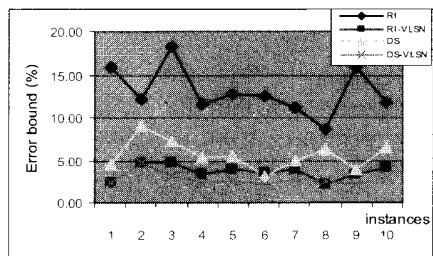


Figure 5. Error bound of solutions from RI, DS, RI-VLSN and DS-VLSN (n=40, m=8)

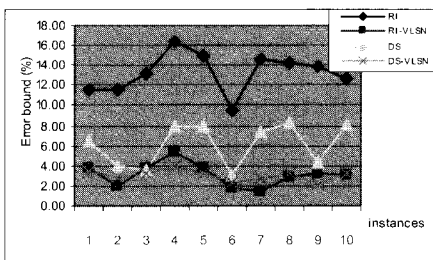


Figure 6. Error bound of solutions from RI, DS, RI-VLSN and DS-VLSN (n=50, m=10)

Table 1. Error with respect to the best value for (n=100, m=20)

Problem (n=100, m=20)	Error (%)			
	RI	DS	RI-VLSN	DS-VLSN
1	11.16	3.60	0.33	0.00
2	9.28	4.59	0.27	0.00
3	12.59	3.50	1.10	0.00
4	14.01	1.7	1.92	0.00
5	14.49	3.32	2.80	0.00
6	10.28	1.93	2.38	0.00
7	10.66	3.63	0.75	0.00
8	13.42	2.22	1.64	0.00
9	11.90	2.98	0.88	0.00
10	12.21	4.49	2.92	0.00
MAX	14.49	4.59	2.92	0.00
AVG	12.00	3.14	1.50	0.00
Time(second)	0.0031	0.008	11.273	7.6281

Table 2. Error with respect to the best value for (n=200, m=40)

Problem (n=200, m=40)	Error (%)			
	RI	DS	RI-VLSN	DS-VLSN
1	13.01	3.73	2.28	0.00
2	14.22	3.15	1.61	0.00
3	11.12	2.63	0.40	0.00
4	16.99	2.55	2.21	0.00
5	15.99	3.65	1.91	0.00
6	13.47	3.34	2.44	0.00
7	13.27	3.00	2.19	0.00
8	12.86	2.51	2.28	0.00
9	14.83	3.82	1.83	0.00
10	14.54	2.51	1.62	0.00
MAX	16.99	3.82	2.44	0.00
AVG	14.03	3.09	1.88	0.00
Time(seconds)	0.0047	0.0406	70.492	27.488

Table 3. Average computational time for the heuristics

Problem Size	Computational Time (seconds)				
	LB	RI	DS	RI-VLSN	DS-VLSN
n=15, m=3	13.5	0	0	0.0205	0.0078
n=30, m=6	849.5	0	0.0016	0.2608	0.0564
n=40, m=8	14207.6	0.0015	0.0016	0.6891	0.1096
n=50, m=10	48672.0	0	0.0060	1.3170	0.2453

5. Conclusion

In this paper, the constructive heuristics and very large scale neighborhood (VLSN) search algorithms are presented in solving an integrated inventory-transportation problem in a deterministic setting. An inbound material-collection system with one warehouse, multiple supplies and multiple items, is considered. The

RI and DS heuristics are implemented to construct initial solutions by partitioning items into several subsets. Then, these solutions are improved by applying the VLSN. The computational results show that the VLSN can efficiently find near-optimal solutions with an average error bound of about 3% for most cases. For future research, the VLSN will be applied to solve other combinatorial optimization problems. Comparison of the VLSN with meta-heuristics is another research direction.

6. References

- [1] Ahuja, R.K., Ergun, O., Orlin, J.B. and Punnen, A.P., A Survey of Very Large-Scale Neighborhood Search Techniques, *Discrete Applied Mathematics*, Vol.123, pp.75-102, 2002.
- [2] Ahuja, R.K., Jha, K.C., Orlin, J. B. and Sharma, D., Very Large Scale Neighborhood Search Algorithm for the Quadratic Assignment Problem, Submitted to *INFORMS Journal on Computing*, 2002.
- [3] Ahuja, R.K. and Orlin, J.B., Very large Scale Neighborhood Search for Airline Fleet Scheduling, *SIAM News*, Vol.35, Number 9, November Issue, 2002.
- [4] Ahuja, R.K., Orlin, J.B. and Sharma, D., Very Large Scale Neighborhood Search, *International Transactions in Operations Research*, Vol.7, pp.301-317, 2000.
- [5] Anily, S., and Federgruen, A., One Warehouse Multiple Retailer Systems with Vehicle Routing Costs, *Management Science*, Vol.36, pp.92-114, 1990.
- [6] Anily, S., The General Multi-retailer EOQ Problem with Vehicle Routing Costs, *European Journal of Operational Research*, Vol.79, pp.451-473, 1994.
- [7] Bertazzi, L., Paletta, G. and Speranza, M.G., Deterministic Order-up-to Level Policies in an Inventory Routing Problem, *Transportation Science*, Vol.36, pp.119-132, 2002.
- [8] Cousineau-Ouimet, K., A Tabu Search Heuristic for the Inventory Routing Problem, *Proceedings of 37th Annual ORSNZ Conference*, 2002.
- [9] Croes, G.A., A Method for Solving Traveling-salesman Problems, *Operations Research*, Vol.6, pp.791-812, 1958.
- [10] Federgruen, A., and Zipkin, P., A Combined Vehicle Routing and Inventory Allocation Problem, *Operations Research*, Vol.32(5), pp.1019-1037, 1984.
- [11] Lin, S., Computer Solutions of the Traveling Salesman Problem, *Bell Systems Technical Journal*, Vol.44, pp.2245-2269, 1965.
- [12] Qu, W.W., Bookbinder, J.H., and Iyogun, P., An Integrated Inventory-transportation System with Modified Periodic Policy for Multiple Products, *European Journal of Operational Research*, Vol.115, pp.254-269, 1999.
- [13] Rosenkrantz, D.J., Stearns, R.E., and Lewis, P.M., An Analysis of Several Heuristics of Traveling Salesman Problem, *SIAM Journal of Computing*, Vol.6, pp.563-581, 1977.
- [14] Sindhuchao, S., An Inventory-Transportation System for Multi-item Joint Replenishment with Limited Vehicle Capacity, Ph.D. Dissertation, Industrial Engineering Department Faculty of Engineering, Chulalongkorn University, 2004
- [15] Viswanathan, S., and Mathur, K., Integrating Routing and Inventory Decisions in One-warehouse Multi-retailer Multi-product Distribution Systems, *Management Science*, Vol.43(3), pp.294-312, 1997.