Performance Measurement for Inventory Routing

Jin-Hwa Song
Martin Savelsbergh

The Logistics Institute
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Abstract

An important reason for companies to introduce vendor managed inventory programs is to eliminate distribution inefficiencies occurring due to the timing of customers' orders. By managing their customers' inventories suppliers may be able to reduce demand variability and therefore distribution costs. We develop technology to measure the effectiveness of distribution strategies. Popular practical performance measures, such as volume delivered per mile, are effective in measuring relative performance, but inadequate to measure absolute performance. We develop a methodology that allows the computation of tight lower bounds on the total mileage required to satisfy customer demand over a period of time. As a result companies will be able to gain insight into the effectiveness of their distribution strategy.

1 Introduction

Vendor managed inventory resupply (VMI) has become a popular strategy to reduce inventory holding and/or distribution costs. In environments where VMI partnerships are in effect, the vendor is allowed to choose the timing and size of deliveries. In exchange for this freedom, the vendor agrees to ensure that its customers do not run out of product. In a more traditional relationship, where customers call in their orders, large inefficiencies can occur due to the timing of customers' orders, i.e., high inventory and high distribution costs. By employing VMI partnerships companies may be able to reduce demand variability and therefore their inventory holding and distribution costs. Realizing the cost savings opportunities of VMI partnerships, however, is not an easy task, particularly with a large number and variety of customers. The inventory routing problem (IRP) seeks to do exactly that: determining a distribution strategy that minimizes long term distribution costs. A large body of literature on the IRP exists; Campbell et al. [3] and Kleywegt et al. [5], among others, contain an overview of the major research activities in this area.

In this paper, we do not focus on developing distribution strategies, but instead on measuring the effectiveness of distribution strategies. A popular performance measure used in practice to evaluate distribution strategies in an environment where VMI partnerships are in effect is the volume delivered per mile or *volume per mile* for short. As the volume that needs to be delivered by the vendor over a given period of time is determined by the total usage of its customers, and not under the control of the vendor, the vendor strives to minimize the total mileage required to deliver product. However, volume per mile by itself is not a meaningful number, because it is impacted by many factors, such as the geography of customer locations and customer usage patterns, but it is valuable for comparing performance in consecutive periods of time. If a company has a stable customer set and customer usage patterns do not fluctuate much, then an increase (decrease) in volume per mile indicates that distribution planning is improving (worsening).

The above discussion shows that volume per mile is a useful measure for monitoring relative distribution strategy performance. However, volume per mile cannot be used to determine, in an absolute sense, the quality of a distribution strategy. We develop a methodology that allows the computation of tight lower bounds on the total mileage required to satisfy customer demand over a period of time (and thus upper bounds on volume per mile). As a result companies will be able to gain insight into the effectiveness of their distribution strategy.

The remainder of the paper is organized as follows. In Section 2 we present a simple bound on the minimum total mileage required to satisfy customer demand. In Section 3 we analyze two 2-customer examples. The analysis reveals a crucial insight that forms the basis for the methodology developed and discussed in Section 4. In Section 5, we present a variety of computational experiments conducted with real-life data to evaluate our proposed methodology. Finally, in Section 6, we discuss other potential uses of the technology developed.

2 A Simple Bound

Consider the following variant of the inventory routing problem. A single product has to be distributed from a single facility to a set I of n customers over a period of time of length T. Each customer $i \in I$ has the capability to maintain a local inventory of product up to a maximum of C_i. In the period of interest customer i consumes an amount u_i of product. A fleet of homogeneous vehicles, with capacity Q, is available for the distribution of the product. We assume an unlimited supply of product and an unlimited number of vehicles in the fleet. We denote the travel distance between two locations i and j by t_{ij}. The objective is to obtain an accurate estimate of the minimum total mileage required to satisfy customer demand. Observe that when $C_i \geq Q \; \forall i \in I$, then the optimal distribution strategy is to always deliver a full truck load to a customer right when the customer's storage tank becomes empty. The resulting total distance is $\sum_{i \in I} \frac{u_i}{Q} 2t_{0i}$, where 0 denotes the plant. Therefore, a simple lower bound on the minimum total mileage required to satisfy customer demand is obtained by simply assuming that all customers' storage capacities are greater than the truck capacity, i.e.,

$$\sum_{i \in I} \frac{u_i}{Q} 2t_{0i}.$$

This results in the following simple upper bound on volume per mile

$$\frac{\sum_{i \in I} u_i}{\sum_{i \in I} \frac{u_i}{Q} 2t_{0i}}.$$

3 Towards an Improved Bound

In practice, deliveries to customers with storage capacity less than the truck's capacity, i.e., $C_i < Q$, are usually combined with other deliveries to ensure a high utilization of the truck's capacity. The analysis of the two 2-customer examples presented below suggests how to incorporate varying storage capacity at customers in the calculation of a lower bound on the minimum total mileage required to satisfy customer demand.

3.1 Example 1

Consider the distribution environment depicted in Figure 1, i.e., a single plant and two customers.

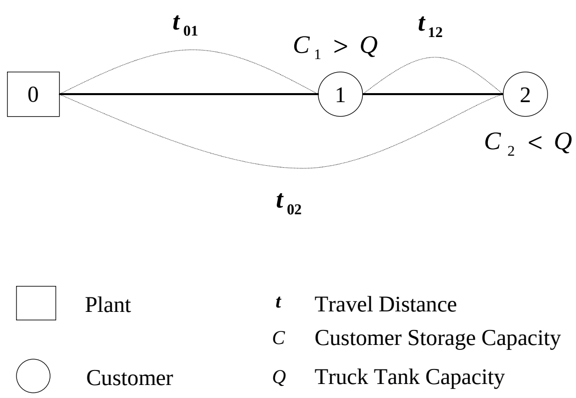

Figure 1: Two customer configuration of Example 1

In this example, $LB_1 = \frac{u_1}{Q} 2t_{01} + \frac{u_2}{Q} 2t_{02}$. Since $C_2 < Q$, whenever a truck goes to Customer 2 with full truck load, $Q - C_2$ of product is left in its tank. Note that $t_{02} = t_{01} + t_{12}$ in this example. The leftover product can be used to satisfy the need for product of Customer 1 at no extra cost. To deliver u_2 to Customer 2, at least $\frac{u_2}{C_2}$ deliveries have to be made. Therefore, at least $\frac{u_2}{C_2}(Q - C_2)$ leftover product is available for Customer 1. Two cases have to be considered: (1) the leftover product is sufficient to satisfy Customer 1's needs, and (2) the leftover product is insufficient to satisfy Customer 1's needs. Let D^* denote the best possible lower bound on the minimum total mileage required to satisfy customer demand.

Case 1 : If $\frac{u_2}{C_2}(Q - C_2) \geq u_1$, then

$$D^* = \frac{u_2}{C_2} 2t_{02}.$$

Case 2 : If $\frac{u_2}{C_2}(Q - C_2) < u_1$, then

$$D^* = \frac{u_2}{C_2} 2t_{02} + \frac{u_1 - \frac{u_2}{C_2}(Q - C_2)}{Q} 2t_{01}.$$

For Case 2, we sent full trucks to Customer 1 to satisfy the remaining product need, i.e., $u_1 - \frac{u_2}{C_2}(Q - C_2)$.

Observation 1. *The delivery patterns used in the best possible lower bound are among* $(Q, 0), (Q - C_2, C_2)$, *and* $(0, C_2)$.

3.2 Example 2

Consider the distribution environment depicted in Figure 2, i.e., a single plant and two customers. In this example, $LB_1 = \frac{u_1}{Q} 2t_{01} + \frac{u_2}{Q} 2t_{02}$. Since $C_1 < Q$, whenever a truck goes to Customer 1 with full truck load, $Q - C_1$ of product is left in its tank. So if this leftover product is used to satisfy product need of Customer 2, $\frac{Q}{Q-C_1}$ trips with leftover product are necessary to deliver Q. Whenever leftover product is delivered to Customer 2, $t_{12} + t_{02} - t_{01}$ additional miles are incurred. Therefore, the travel distance incurred to deliver Q to Customer 2 with leftover product from Customer 1 is $\frac{Q}{Q-C_1}(t_{12} + t_{02} - t_{01})$. The travel distance incurred to deliver Q to Customer 2 directly from the plant is $2t_{02}$. Consequently, if $\frac{Q}{Q-C_1}(t_{12} + t_{02} - t_{01}) < 2t_{02}$, it is better to use leftover product at Customer 1 to satisfy the product need of Customer 2. For the remainder, assume that this is the case, i.e., $\frac{Q}{Q-C_1}(t_{12} + t_{02} - t_{01}) < 2t_{02}$. Two cases have to be considered: (1) the leftover product is sufficient to satisfy Customer 2's needs, and (2) the leftover product is insufficient to satisfy Customer 2's needs.

Case 1 : If $\frac{u_1}{C_1}(Q - C_1) \geq u_2$, then

$$D^* = \frac{u_2}{Q - C_1}(t_{01} + t_{12} + t_{02}) + \frac{u_1 - \frac{u_2}{Q-C_1} C_1}{C_1} 2t_{01}$$

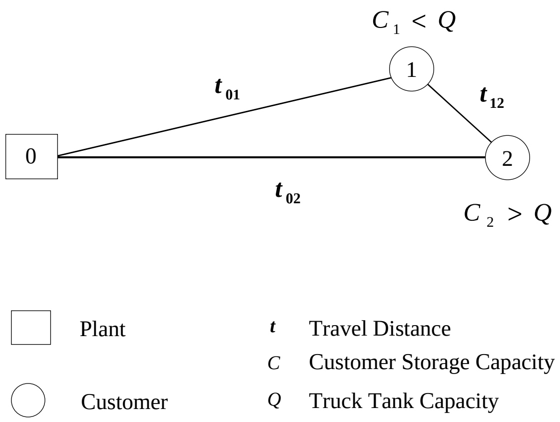

Figure 2: Two customer configuration of Example 2

Case 2 : $\frac{u_1}{C_1}(Q - C_1) < u_2$, then

$$D^* = \frac{u_1}{C_1}(t_{01} + t_{12} + t_{02}) + \frac{u_2 - \frac{u_1}{C_1}(Q - C_1)}{Q} 2t_{02}.$$

Observation 2. *The only delivery patterns used in an optimal solution are among* $(C_1, 0)$, $(0, Q)$, *and* $(C_1, Q - C_1)$.

The two observations above form the basis for the methodology developed to compute improved bounds on the minimum total mileage required to satisfy customer demand.

4 An Improved Bound

Define a feasible delivery pattern $P_j = (d_{j1}, d_{j2}, ..., d_{jn})$ to be a delivery pattern that satisfies $\sum_{i \in I} d_{ji} \leq Q$ and $0 \leq d_{ji} \leq C_i \; \forall i \in I$. Let $\delta(P_j) = \{i \in I : d_{ji} > 0\}$ denote the set of customers visited in delivery pattern P_j. The cost of delivery pattern P_j, denoted as $c(P_j)$, is the value of an optimal solution to the traveling salesman problem involving the plant and the customers in $\delta(P_j)$. Let $\mathcal{P}$ be the set of all feasible delivery patterns and let x_j be a decision variable indicating how many times delivery pattern P_j is used. Then the optimal objective function value of the following linear program, called the pattern selection LP, provides a lower bound on the total mileage required to satisfy customer demand

$$
\begin{aligned}
D^* = \quad & min \quad \sum_{j:P_j \in \mathcal{P}} c(P_j)x_j \\
& s.t. \quad \sum_{j:P_j \in \mathcal{P}} d_{ji}x_j \geq u_i, \quad \forall i \in I \\
& \qquad\quad x_j \geq 0
\end{aligned}
$$

There are two major obstacles to using this linear program:

- The number of feasible delivery patterns is prohibitively large.
- The calculation of the cost of each delivery pattern involves the solution of a traveling salesman problem.

In the remainder of this section we discuss how these obstacles can be handled in practice. (We will assume throughout that distances satisfy the triangle inequality.)

We will start by showing that a much smaller set of delivery patterns can be considered when solving the linear program (an insight resulting from the analysis presented in the previous section).

Definition 1. *(Base Pattern) A feasible delivery pattern P is a base pattern if at most one customer, say k, in $\delta(P)$ receives a delivery quantity less than $min(C_k, Q)$, and, in that case, the delivery quantity is $Q - \sum_{i \in \delta(P) \setminus \{k\}} C_i$.*

The base patterns can be divided into two classes:

1. $\sum_{i \in \delta(P)} C_i \leq Q$ so that $d_i = C_i \; \forall i \in \delta(P)$, and
2. $\sum_{i \in \delta(P)} C_i > Q$ so that there exists one customer, say k, with $d_k = Q - \sum_{i \in \delta(P) \setminus \{k\}} C_i$ and $d_i = min(C_i, Q) \; \forall i \in \delta(P) \setminus \{k\}$.

Theorem 1. *The base patterns are sufficient to find an optimal solution to the Pattern Selection LP.*

Proof. Case 1. A feasible pattern P with $\sum_{i \in \delta(P)} d_i < Q$.
Suppose $\sum_{i \in \delta(P)} C_i < Q$. Then there exits a base pattern P' with $\delta(P) = \delta(P')$ and $d'_i = C_i \; \forall i \in \delta(P)$. Because $c(P) = c(P')$ and $d_i \leq d'_i \; \forall i \in \delta(P)$, we can replace P in any optimal solution by P'. Suppose $\sum_{i \in \delta(P)} C_i > Q$. Then there exits a pattern P' with $\delta(P) = \delta(P')$, $d'_i \geq d_i; \forall i \in \delta(P)$, and $\sum_{i \in \delta(P')} d'_i = Q$. Because $c(P) = c(P')$ and $d_i \leq d'_i \; \forall i \in \delta(P)$, we can replace P in any optimal solution by P'. Such patterns are covered in Case 2.

Case 2. A feasible pattern P with $\sum_{i \in \delta(P)} d_i = Q$.
We will show that such a pattern can be represented by a convex combination of base patterns with $\delta(\cdot) \subseteq \delta(P)$ (which implies that $c(\cdot) \leq c(P)$). Consider a feasible pattern P with $\sum_{i \in \delta(P)} d_i = Q$. Without loss of generality, assume that $\delta(P) = \{1, 2, ..., m\}$. Let $\{P_1, P_2, ..., P_{n_p}\}$ be the set of base patterns with $\delta(\cdot) \subseteq \delta(P)$. Let A be the $(m+1) \times n_p$ matrix in which the j^{th} column is $(d_{j1}, d_{j2}, ..., d_{jm}, 1)^T$. (Where d_{ji} is pattern P_j's i^{th} element.) Let $b^T = (d_1, d_2, ..., d_m, 1)$. We have to show that the linear system $Ax = b,\ x \geq 0$ has a feasible solution. We do so using Farkas' Lemma.

Farkas' Lemma. *A linear system $Ax = b,\ x \geq 0$ has a feasible solution if and only if $yb \geq 0$ for each y with $yA \geq 0$.*

Suppose $yb < 0$ for some y with $yA \geq 0$. Without loss of generality, we assume $y_1 \geq y_2 \geq \cdots \geq y_m$. Then there exists a base pattern P' such that $d'_i = min(C_i, Q)\ \forall i \in \{l+1, l+2, ..., m\}$, $d'_l = Q - \sum_{i=l+1}^{m} min(C_i, Q)$, and $d'_i = 0,\ \forall i \in \{1, 2, ..., l-1\}$. Since $(d'_1, d'_2, ..., d'_m, 1)^T$ is a column of A, we have $\sum_{i=1}^{m} d'_i y_i + y_{m+1} \geq 0$. Now, since $\sum_{i=1}^{m} d_i = \sum_{i=1}^{m} d'_i = Q$ and P is a feasible pattern, $\sum_{i=k}^{m} d_i \leq \sum_{i=k}^{m} d'_i\ \forall k \in \{1, 2, ..., m\}$. Therefore, $\sum_{i=1}^{k} d_i \geq \sum_{i=1}^{k} d'_i\ \forall k \in \{1, 2, ..., m\}$ and thus $\sum_{i=1}^{k}(d_i - d'_i) \geq 0\ \forall k \in \{1, 2, ..., m-1\}$ and $\sum_{i=1}^{m}(d_i - d'_i) = 0$. Since $y_i \geq y_{i+1}$, we have $\sum_{i=1}^{k}(d_i - d'_i)y_k \geq \sum_{i=1}^{k}(d_i - d'_i)y_{k+1}\ \forall k \in \{1, 2, ..., m-1\}$. Since $\sum_{i=1}^{m}(d_i - d'_i) = 0$, we have $\sum_{i=1}^{m}(d_i - d'_i)y_m = 0$. By summing these m inequalities, we obtain $\sum_{i=1}^{m}(d_i - d'_i)y_i \geq 0$. But then $\sum_{i=1}^{m} d'_i y_i + y_{m+1} \leq \sum_{i=1}^{m} d_i y_i + y_{m+1} = yb < 0$, a contradiction. □

Now that we have significantly reduced the number of delivery patterns, we turn our attention to the number of customers visited in a delivery pattern as that impacts the effort required to compute the cost of a delivery pattern.

For any natural number k, let $C'_i = \frac{Q}{k}$ if $C_i < \frac{Q}{k}$ and $C'_i = C_i$ if $C_i \geq \frac{Q}{k}$. Observe that with these modified storage capacities a base pattern contains at most k customers. Let LB_k denote the optimal value of the pattern selection LP with base patterns based on the modified storage capacities. It is easy to see that LB_k provides a lower bound on D^* for every k and that $LB_1 \leq LB_2 \leq LB_3 \leq ....$ Finally, when $\frac{Q}{k} \leq min\{C_1, C_2, ..., C_n\}$, then $LB_k = D^*$. Note that the simple bound discussed in Section 2 is equal to LB_1.

For any natural number k, we can also compute an upper bound UB_k on D^* as follows. We let UB_k be the optimal objective function value of the pattern selection LP in which we only consider base patterns with at most k customers. It is easy to see that $UB_1 \geq UB_2 \geq UB_3 \geq ...$ and that when $k \geq \left\lceil \frac{Q}{min\{C_1, C_2, ..., C_n\}} \right\rceil$, then $UB_k = D^*$.

Our computational experiments have shown that tight bounds on D^* are already obtained for values $k = 3$ and $k = 4$, in the sense that the gap between LB_k and UB_k is very small (for the data sets of interest to us). Furthermore, for values $k = 3$ and $k = 4$, the traveling salesman problems that have to be solved involve at most 4 and 5 cities, respectively, and thus can be solved relatively easily by enumeration.

Our initial computational experiments have also shown that even though we have significantly reduced the number of delivery patterns in the pattern selection LP by restricting ourselves to base patterns, as the number of customers increases - especially the number of customers whose storage capacities are small - the number of base patterns increases rapidly. For example, for one of our larger instances with 194 customers 22,575,528 base patterns were generated to compute UB_4. Even when using carefully designed memory-efficient implementations, the memory requirements become excessive. To be able to handle such large instances (and even larger ones) effectively, we have developed two additional techniques.

So far, we have only exploited feasibility considerations to reduce the set of delivery patterns that need to be considered. Next, we will show how optimality considerations can be exploited effectively to reduce the set of delivery patterns that need to be considered. Consider a base pattern $P = \{d_1, d_2, \ldots, d_n\}$, all base patterns P_j visiting a subset of the customers in $\delta(P)$, and the following linear program, called the dominance LP,

$$\begin{aligned} z = \quad & min \sum_{\{j:\delta(P_j)\subsetneq\delta(P)\}} c(P_j)\lambda_j \\ & s.t. \sum_{\{j:\delta(P_j)\subsetneq\delta(P)\}} d_{ji}\lambda_j \geq d_i, \quad \forall i \in \delta(P) \\ & \lambda_j \geq 0 \end{aligned}$$

If $z \leq c(P)$, then the base patterns with $\lambda_j > 0$ collectively dominate the base pattern P and base pattern P can be eliminated from the pattern selection LP. However, even though the size of a dominance LP is small, setting up and solving it for every base pattern to determine if the base pattern is dominated is computationally prohibitive. Therefore, we rely on easily computable upper bounds on the optimal value of a dominance LP for dominance testing; if $z \leq z_{UB} \leq c(P)$, where z_{UB} denotes an upper bound on z, then the base pattern P is dominated and can be eliminated. We compute upper bound z_{UB} by restricting our attention to carefully selected subsets of patterns.

To illustrate, consider a base pattern $P = \{C_1, C_2, C_3, d_4\}$ with $d_4 < C_4$. (Note that this implies that $C_1+C_2+C_3+d_4 = Q$.) If one of the following three conditions is satisfied for P, then P is dominated:

Condition 1: $c(P) \geq c(P_{123}) + \frac{d_4}{min\{Q,C_4\}} c(P_4)$ where $P_{123} = \{C_1, C_2, C_3, 0\}$ and $P_4 = \{0, 0, 0, min\{Q, C_4\}\}$.

Condition 2: $c(P) \geq \left(1 - \frac{d_4}{d_{14}+d_{24}+d_{34}}\right) c(P_{123}) + \frac{d_4}{d_{14}+d_{24}+d_{34}} \left(c(P_{14}) + c(P_{24}) + c(P_{34})\right)$

where $d_{14} = min\{Q - C_1, C_4\}$, $d_{24} = min\{Q - C_2, C_4\}$, $d_{34} = min\{Q - C_3, C_4\}$, $P_{123} = \{C_1, C_2, C_3, 0\}$, $P_{14} = \{C_1, 0, 0, d_{14}\}$, $P_{24} = \{0, C_2, 0, d_{24}\}$, and $P_{34} = \{0, 0, C_3, d_{34}\}$.

Condition 3: $c(P) \geq \left(1 - \frac{2d_4}{d_{124}+d_{134}+d_{234}}\right) c(P_{123}) + \frac{d_4}{d_{124}+d_{134}+d_{234}} \left(c(P_{124}) + c(P_{134}) + c(P_{234})\right)$ where $d_{124} = min\{Q - C_1 - C_2, C_4\}$, $d_{134} = min\{Q - C_1 - C_3, C_4\}$, $d_{234} = min\{Q - C_2 - C_3, C_4\}$, $P_{123} = \{C_1, C_2, C_3, 0\}$, $P_{124} = \{C_1, C_2, 0, d_{124}\}$, $P_{134} = \{C_1, 0, C_3, d_{134}\}$, and $P_{234} = \{0, C_2, C_3, d_{234}\}$.

The effectiveness of these simple dominance tests is demonstrated by the results presented in Table 1. The table shows the number of base patterns before and after applying the dominance tests.

Table 1: Effect of dominance tests

Instance	n	before	after
1	136	5,015,046	3,029,980
2	106	4,488,526	3,395,575
3	157	7,665,722	4,336,466
4	129	7,579,201	5,381,540
5	169	9,086,385	5,420,907
6	147	15,180,701	8,838,137
7	157	14,471,228	8,975,615
8	194	22,575,528	16,640,122

Next, we observe that a pattern selection LP has a large aspect ratio, i.e., a large ratio of number of columns to number of rows. Linear programs with large aspect ratios occur frequently when set partition or set covering formulations are used to model practical situations, for example in air crew scheduling applications. Specialized linear programming solvers exploiting the fact that most variables will have a zero value in an optimal solution have been developed for such problems. In the CPLEX linear optimization system, the specialized linear programming solver for high aspect ratio linear programs is the sifting optimizer. The sifting optimizer solves an LP with only a subset of the variables (assuming a zero solution value for each of the remaining variables). From the solution to this partial LP, the reduced costs of the remaining variables can be computed. Variables with reduced costs less than zero are added to the partial LP, the partial LP is resolved, and the process repeats. If no negative reduced cost variables exist, then the current solution is an optimal solution to the full problem. (This approach was first introduced by IBM under the name SPRINT approach [1].) Table 2 shows a comparison of cpu times for the CPLEX linear optimizer and the CPLEX sifting optimizer on some of the larger instances of the pattern selection LP.

Table 2: Effect of sifting optimizer

Instance	n	# of patterns	# of iterations	default(sec)	sifting(sec)
1	136	3,029,980	5	85.66	77.09
2	106	3,395,575	5	167.15	94.07
3	157	4,336,466	6	118.37	121.34
4	129	5,381,540	6	308.44	244.90
5	169	5,420,907	6	155.29	152.56
6	147	8,838,137	5	360.20	240.83
7	157	8,975,615	6	397.33	254.81
8	194	16,640,122	6	675.98	533.56

In Table 2, we present the number of customers in the instance (n), the number of patterns generated, the number of major iterations of the sifting optimizer, and the cpu time taken by the default and sifting optimizer.

Using CPLEX' sifting optimizer does not resolve our memory issues as all delivery patterns still need to be loaded into memory. Therefore, we have developed our own implementation of a sifting optimizer. During the pattern generation phase, we do not load all base patterns into the linear programming solver, but only a subset of patterns that fits into memory and is highly likely to include an optimal solution. After solving this partial LP, we execute the pattern generation phase again, but this time we evaluate the reduced costs of the patterns (as opposed to their regular cost) and add patterns with a negative reduced cost to the partial LP. The algorithm terminates when no patterns can be added in an iteration.

In implementing this approach, we have to take into account that our pattern generation phase is computationally intensive as it involves solving a, albeit small, TSP for every pattern. As a consequence, we have to strike a proper balance between pattern generation and memory management. The ultimate goal is to solve the pattern selection LP with only two passes through pattern generation, i.e., one to generate a partial pattern selection LP and one to verify that all patterns left out of the partial pattern selection LP have nonnegative reduced costs. We want to avoid having to go through more than two pattern generation passes. To achieve this goal, we generate patterns in a specific order, first patterns involving a stop at a single customer, second patterns involving stops at two customers, third patterns involving stops at three customers, and finally patterns involving stops at four customers, and add patterns to the partial pattern selection LP in batches, solving the partial selection LP after each batch of patterns has been added. A more detailed description can be found in Algorithm 1.

Observe that all patterns of size 1 and size 2 are part of the partial selection LP and that when generating patterns of size 3 and size 4 we always evaluate their reduced cost.

Algorithm 1 Sifting Optimizer

```
generate := true
ρ3 := ...
ρ4 := ...
Generate all patterns of size 1 and 2 and add them to the partial pattern selection LP
Solve the partial pattern selection LP
while generate = true do
  generate := false
  Generate all patterns of size 3 and add those with reduced cost less than ρ3 to the
  partial pattern selection LP
  ρ3 = 0
  if patterns were added to partial selection LP then
    generate := true
    Solve the partial pattern selection LP
  end if
  Generate all patterns of size 4 and add those with reduced cost less than ρ4 to the
  partial pattern selection LP
  ρ4 = 0
  if patterns were added to partial selection LP then
    generate := true
    Solve the partial pattern selection LP
  end if
end while
```

Observe too that we do not only add patterns with a negative reduced cost when we generate patterns of size 3 and size 4. As one of our primary goals is to limit the number of pattern generation passes, we do not want to be too conservative. Ideally, the values ρ_3 and ρ_4 are set based on an analysis of the instance that needs to be solved. However, we have been unable to develop methodology to do so and have take a more pragmatic approach. First, because we have access to a machine with 16Gb of memory the number of patterns of size 3 does not pose a problem and we have used $\rho_3 = \infty$, which means we add all patterns of size 3. Second, after experimenting with a few instances and a few values for ρ_4, we found that $\rho_4 = 20$ performed well. In fact, with $\rho_4 = 20$, all instances were solved in two pattern generation passes, i.e., one pass to generate the partial instance, and one pass to verify optimality of the solution produced. The effect of only incorporating patterns of size 4 with a reduced cost of less than or equal to 20 ($= \rho_4$) is dramatic as can be seen in Table 3. The number of patterns generated is reduced by a factor of more than 20.

Table 3: Number of patterns

Instances	# of 1-stop patterns	# of 2-stop patterns	# of 3-stop patterns	# of 4-stop patterns	# of 4-stop patterns with r.c $\leq \rho_4$
1	136	7,163	220,949	2,801,732	32,213
2	106	5,377	154,719	3,235,373	82,964
3	157	8,225	279,217	4,048,867	75,466
4	129	7,464	284,176	5,089,771	240,997
5	169	9,015	312,962	5,098,761	44,265
6	147	8,414	331,629	8,497,947	36,152
7	157	10,281	422,653	8,542,524	114,402
8	194	16,356	784,857	15,838,715	276,416

5 Computational Experiments

The research reported in this paper was motivated by our long-time collaboration with Praxair a producer and distributor of industrial gases. Praxair used the simple bound discussed in Section 2 to get an idea of the performance of their distribution strategy and to get a sense of the potential savings if additional resources and efforts were invested in improving their distribution strategy. As mentioned, this simple bound has two main deficiencies: it ignores the different storage capacities at customers and it assumes that deliveries can be perfectly timed. Our work eliminates the first deficiency.

We conducted various computational experiments to analyze the effect on the lower bound on the minimum total mileage required to satisfy demand of explicitly taking varying storage capacities into account. The data used in our experiments had usage information for 1974 customers served from 36 plants (with the smallest plant serving only 6 customers and the largest plant serving 147 customers). Each customer is supplied from one particular plant. Consequently, we are dealing with independent 36 instances.

The primary experiment involved computing increasingly tighter lower and upper bounds on D^* the bound on the minimum total mileage required to satisfy demand. The results are displayed in Figure 3.

First, the results shows that limiting ourselves to patterns with at most three or four customers is sufficient to obtain tight bounds on D^*. Second, the results show that allowing more deliveries per trip has a substantial effect on the upper bound, but hardly any effect on the lower bound. The latter result was somewhat counter to our expectations, but has important implications because it suggests that investing in larger storage facilities at customers, which is often discussed as a potential way of reducing distribution costs, may not deliver the desired savings. Finally, by comparing the actual incurred mileage to

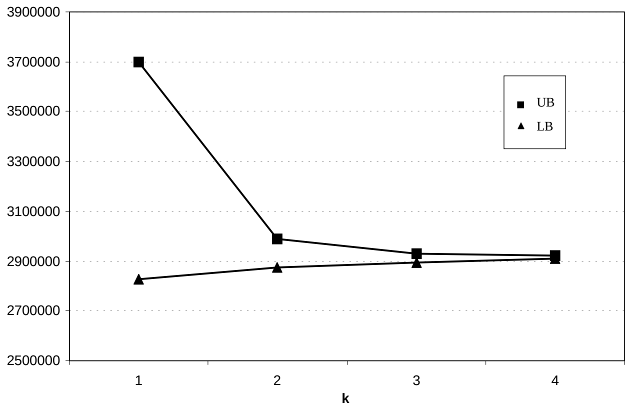

Figure 3: Lower and upper bounds on D^*

LB_4 over a period of time, Praxair will gain insight in the effectiveness of its distribution strategy and in the potential savings that may result from improvements to its distribution strategy.

Next, we investigate whether the behavior observed for the complete system is also observed at the individual plant level. Table 4 shows the number of customers served (n), LB_3, UB_3, their relative gap ($\frac{UB_3-LB_3}{UB_3} \times 100$), LB_4, UB_4, and their relative gap ($\frac{UB_4-LB_4}{UB_4} \times 100$) for each plant.

We see that for bounds LB_4 and UB_4 the largest relative gap is 2.53% for Plant 18 and the smallest relative gap is 0.02% from Plant 1. To understand the cause of the differences, we examined these plants in more detail. Two factors clearly impact the difference between the value of LB_4 and UB_4:

- The number of customers with $C_i < \frac{Q}{4}$
- The number of times we have to make deliveries to customers with $C_i < \frac{Q}{4}$

Note that when all customers served by a plant have $C_i \geq \frac{Q}{4}$, then we have $LB_4 = UB_4$. When we look more closely at Plant 1, we see that when a direct delivery policy would be employed, the number of deliveries is 568.4 (computed as $\sum_i \frac{u_i}{min(C_i,Q)}$). Among these 554.0 correspond to deliveries to customers with $C_i \geq \frac{Q}{4}$, i.e., 97.5% of the total number if deliveries. On the other hand, for Plant 18 the number of deliveries is 163.8 when a direct delivery policy is employed, out of which 100.9 correspond to deliveries to customers with $C_i \geq \frac{Q}{4}$, i.e., only 61.6% of the total number of deliveries.

plant	n	LB_3	UB_3	% Gap	LB_4	UB_4	% Gap
1	28	53,079.7	53,209.2	0.24%	53,198.8	53,208.9	0.02%
2	18	11,483.5	11,498.6	0.13%	11,494.9	11,498.6	0.03%
3	67	80,861.5	82,718.7	2.25%	81,332.0	82,424.1	1.32%
4	64	117,050.0	118,072.0	0.87%	117,610.0	117,931.0	0.27%
5	70	16,078.8	16,729.7	3.89%	16,283.9	16,502.8	1.33%
6	45	18,000.0	18,337.9	1.84%	18,062.1	18,139.1	0.42%
7	46	63,249.4	64,216.0	1.51%	63,698.3	64,001.4	0.47%
8	83	171,268.0	174,944.0	2.10%	172,768.0	174,409.0	0.94%
9	56	100,915.0	101,528.0	0.60%	101,260.0	101,361.0	0.10%
10	70	100,833.0	102,448.0	1.58%	101,505.0	102,295.0	0.77%
11	40	63,482.5	63,996.1	0.80%	63,723.9	63,961.2	0.37%
12	69	63,416.3	64,038.5	0.97%	63,665.0	63,975.0	0.48%
13	46	151,401.0	152,860.0	0.95%	151,886.0	152,704.0	0.54%
14	67	28,684.7	29,472.3	2.67%	28,940.6	29,281.8	1.17%
15	74	204,361.0	204,673.0	0.15%	204,486.0	204,601.0	0.06%
16	39	71,920.5	72,374.0	0.63%	72,174.7	72,314.0	0.19%
17	6	8,760.3	8,846.6	0.98%	8,826.9	8,846.6	0.22%
18	29	17,123.8	19,038.0	10.05%	18,094.2	18,563.6	2.53%
19	47	191,068.0	192,031.0	0.50%	191,409.0	191,923.0	0.27%
20	64	108,263.0	108,416.0	0.14%	108,297.0	108,415.0	0.11%
21	75	39,199.2	39,632.1	1.09%	39,514.8	39,605.9	0.23%
22	40	65,636.0	65,792.7	0.24%	65,695.9	65,778.6	0.13%
23	37	15,468.1	15,923.4	2.86%	15,597.4	15,810.4	1.35%
24	75	75,980.6	76,768.4	1.03%	76,351.1	76,656.8	0.40%
25	29	39,779.3	40,096.2	0.79%	39,878.1	40,096.2	0.54%
26	43	122,804.0	123,867.0	0.86%	123,168.0	123,820.0	0.53%
27	101	158,575.0	163,854.0	3.22%	160,117.0	162,432.0	1.43%
28	54	30,963.8	31,301.4	1.08%	31,028.8	31,201.8	0.55%
29	106	87,907.8	90,052.0	2.38%	88,336.9	88,984.8	0.73%
30	86	162,410.0	163,948.0	0.94%	162,965.0	163,404.0	0.27%
31	81	104,428.0	105,404.0	0.93%	104,829.0	105,077.0	0.24%
32	147	183,515.0	186,359.0	1.53%	184,190.0	185,407.0	0.66%
33	11	56,356.1	56,552.5	0.35%	56,475.9	56,552.5	0.14%
34	23	69,516.8	69,575.8	0.08%	69,544.6	69,575.8	0.04%
35	22	16,934.1	17,157.5	1.30%	17,064.2	17,157.5	0.54%
36	16	25,441.5	25,534.2	0.36%	25,491.7	25,534.2	0.17%
Overall	1974	2,896,215.3	2,931,265.8	1.20%	2,908,964.7	2,923,450.6	0.50%

Table 4: Lower and upper bounds for $k = 3, 4$

Finally, we examine the improvements in the lower bounds LB_k at the individual plant level. Table 5 shows the number of customers served (n), LB_1, LB_2, LB_3, LB_4, and the percentage increase between LB_1 and LB_4 ($\frac{LB_4 - LB_1}{LB_1} \times 100$) for each plant.

plant	n	LB_1	LB_2	LB_3	LB_4	% increase
1	28	51,752.8	52,821.3	53,079.7	53,198.8	2.79%
2	18	8,999.7	10,947.1	11,483.5	11,494.9	27.73%
3	67	77,230.1	80,252.8	80,861.5	81,332.0	5.31%
4	64	113,659.0	115,969.0	117,050.0	117,610.0	3.48%
5	70	15,058.4	15,842.1	16,078.8	16,283.9	8.14%
6	45	17,324.8	17,815.5	18,000.0	18,062.1	4.26%
7	46	61,648.3	62,772.6	63,249.4	63,698.3	3.33%
8	83	166,972.0	169,967.0	171,268.0	172,768.0	3.47%
9	56	99,193.7	100,562.0	100,915.0	101,260.0	2.08%
10	70	98,476.9	100,058.0	100,833.0	101,505.0	3.07%
11	40	62,163.9	63,107.5	63,482.5	63,723.9	2.51%
12	69	59,435.1	62,013.0	63,416.3	63,665.0	7.12%
13	46	149,773.0	150,768.0	151,401.0	151,886.0	1.41%
14	67	27,336.5	28,298.2	28,684.7	28,940.6	5.87%
15	74	202,434.0	203,881.0	204,361.0	204,486.0	1.01%
16	39	70,090.0	71,377.4	71,920.5	72,174.7	2.97%
17	6	8,619.3	8,693.8	8,760.3	8,826.9	2.41%
18	29	15,456.3	16,337.5	17,123.8	18,094.2	17.07%
19	47	189,761.0	190,547.0	191,068.0	191,409.0	0.87%
20	64	106,867.0	108,059.0	108,263.0	108,297.0	1.34%
21	75	36,984.9	38,662.9	39,199.2	39,514.8	6.84%
22	40	64,681.1	65,546.7	65,636.0	65,695.9	1.57%
23	37	12,230.2	14,469.9	15,468.1	15,597.4	27.53%
24	75	73,946.8	75,521.2	75,980.6	76,351.1	3.25%
25	29	39,151.8	39,646.9	39,779.3	39,878.1	1.86%
26	43	121,219.0	122,388.0	122,804.0	123,168.0	1.61%
27	101	153,825.0	156,635.0	158,575.0	160,117.0	4.09%
28	54	29,295.6	30,450.1	30,963.8	31,028.8	5.92%
29	106	84,495.6	86,928.1	87,907.8	88,336.9	4.55%
30	86	159,442.0	161,442.0	162,410.0	162,965.0	2.21%
31	81	102,371.0	103,798.0	104,428.0	104,829.0	2.40%
32	147	180,813.0	182,331.0	183,515.0	184,190.0	1.87%
33	11	56,119.8	56,217.7	56,356.1	56,475.9	0.63%
34	23	69,119.6	69,458.9	69,516.8	69,544.6	0.61%
35	22	16,364.3	16,824.4	16,934.1	17,064.2	4.28%
36	16	24,933.7	25,333.3	25,441.5	25,491.7	2.24%
overall	1974	2,827,245.2	2,875,743.9	2,896,215.3	2,908,964.7	2.89%

Table 5: Lower bounds increases

The largest percentage increase is 27.73% for Plant 2 and the smallest percentage increase is 0.61% for Plant 34. Again, we can explain this difference by analyzing what happens when a direct delivery policy is employed. When a direct delivery policy is employed, Plant 2 has to make 52.8 deliveries. Among these 46.8 are to customers with $C_i \geq \frac{Q}{4}$ (with 2.7 to customers with $C_i \geq Q$), i.e., 88.6% (5.1%) of the total number of deliveries. On the other hand, Plant 34 has to make 420.4 when a direct delivery policy is employed, out of which 409.6 deliveries are to customers with $C_i \geq \frac{Q}{4}$ (with 350.0 to customers with $C_i \geq Q$), i.e., 97.4 % (83.3%) of the total number of deliveries.

The system wide percentage increase is only 2.89%. There are two reasons why the increase in the value of the lower bound is relatively small. First, 5,957,872,921 ft^3 out of a total of 8,150,328,576 ft^3, or 73.1%, has to be delivered to customers with $C_i \geq Q$. Second, the geography of customers with $C_i < Q$ is such that they can be combined into delivery trips that do not increase the total mileage by much (the pattern selection LP will identify optimal combinations of customers).

6 Other Uses of the Technology

In addition to providing a lower bound on the total mileage required to satisfy demand, the technology may have other benefits. For example, the selected base patterns may suggest effective practical delivery trips.

We demonstrate this potential benefit by examining a small instance introduced by Fisher et al. [4, 2] to illustrate the complexity of inventory routing problems.

Example. Consider the instance depicted Figure 4. The vehicle capacity is 5000 and

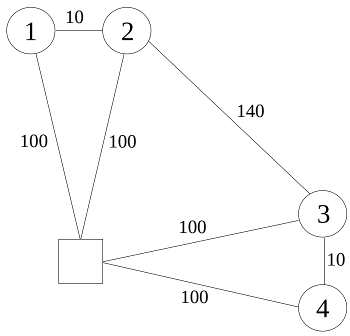

Figure 4: An example with 4 customers.

customer tank capacity and usage data is as follows:

Customer	C_i	u_i
1	5000	1000
2	3000	3000
3	2000	2000
4	4000	1500

The relevant optimal tour costs can be derived from the network shown, e.g., the optimal tour costs for visiting customers 1 and 2, denoted by c_{12}, is equal to \$210. A simple schedule jointly replenishes customers 1 and 2 as well as customers 3 and 4 on a daily basis. This schedule is natural because 1 and 2 (3 and 4, respectively) are near each other. Each customer i receives a quantity equal to its daily consumption u_i. The long-run average cost of this schedule is 420 miles per day. An improved schedule consists of a cycle that repeats every two days. On the first day, one trip is taken that replenishes 3000 gallons to 2 and 2000 gallons to 3, at a cost of 340 miles. On the second day, two trips are taken. The first trip replenishes 2000 gallons to 1 and 3000 gallons to 2. The second trip replenishes 2000 gallons to 3 and 3000 gallons to 4. Each trip costs 210 miles. The long-run time average cost of this schedule is 380 miles per day, which is nearly 10% lower than the first schedule. Fisher et al. observe that though it is easy to verify that the second schedule is better than the first, it is not at all obvious how to derive the second schedule.

By assuming a time period of a single day, the following pattern selection LP will be constructed, where we have left out routes visiting customer 1 and 3, 2 and 4, and 1 and 4, as no distance information is provided that allows the calculation of the tour length:

$$
\begin{array}{lllllllllll}
min & 200x_1 & +200x_2 & +200x_3 & +200x_4 & +210x_5 & +340x_6 & +210x_7 & +210x_8 & & \\
s.t. & 5000x_1 & & & & +2000x_5 & & & & \geq & 1000 \\
 & & 3000x_2 & & & +3000x_5 & +3000x_6 & & & \geq & 3000 \\
 & & & 2000x_3 & & & +2000x_6 & +2000x_7 & +1000x_8 & \geq & 2000 \\
 & & & & 4000x_4 & & & +3000x_7 & +4000x_8 & \geq & 1500 \\
 & x \geq 0 & & & & & & & & &
\end{array}
$$

The optimal solution selects patterns 5, 6, and 7 with value 0.5, which, given a time period of a single day, can be interpreted as using these patterns every other day. This corresponds precisely to the improved solution presented by Fisher et al. Our approach also shows that no better solution exists!

The above example illustrates that the technology can be used for purposes other than performance measurement. The solution may suggest routing patterns that have not been considered so far. The technology may also be used to assist in tactical and strategic decisions. For example, it may be used to evaluate capital investment decisions related to increasing storage capacity at customers, or it may be used to evaluate customer - plant alignments in situations with multiple production facilities servicing the set of customers.

Acknowledgement

We want to express our thanks to Laurie Dutton from Praxair for many valuable discussions.

References

[1] R. Anbil, R. Tanga, and E.L. Johnson. A global optimization approach to crew scheduling. *IBM Systems Journal*, 31:71–78, 1991.

[2] W. Bell, L. Dalberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.

[3] A. M. Campbell, L. Clarke, A. Kleywegt, and M. W. P. Savelsbergh. Inventory routing. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*. Kluwer Academic Publishers, 1998.

[4] M. Fisher, A. Greenfield, R. Jaikumar, and P. Kedia. Real-time scheduling of a bulk delivery fleet: Practical application of lagrangean relaxation. Technical report, The Wharton School, University of Pennsylvania, Department of Decision Sciences, October 1982.

[5] A. Kleywegt, V. Nori, and M. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36:94–118, 2002.