December 8, 1998
RT0283
Computer Science 13 pages

Research Report

Optimizing the distribution cost of a cooking oil company by balancing direct/indirect deliveries

Toshiyuki Hama

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Optimizing the distribution cost of a cooking oil company by balancing direct/indirect deliveries

Toshiyuki Hama
IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.
e-mail: hama@jp.ibm.com

Abstract

We have considered an optimization problem of the distribution cost of a cooking oil company. They have two alternative delivery methods, direct/indirect deliveries, to a demand region near their plant. Since the shipping charges of the two delivery methods have quite different cost structures, we have to appropriately balancing total demand between the two methods. We modeled a primal part of the problem as a multi-attribute multi-sized bin packing problem and solved by several heuristic search methods. Results are compared to the lower bounds obtained by LP relaxation.

1 Introduction

To keep a company competitive in a recent market place, optimizing material flow through the company's supply chain has been recognized very important. In the case of commodity makers, the distribution cost occupies fairly large part of the total cost of a final product. Therefore we can effectively improve the competitiveness of such a company by optimizing the distribution cost.

We have considered an optimization problem of minimizing the distribution cost of a Japanese cooking oil company. The company has two plants in Japan: one of them covers demands from west and south Japan, and the other covers demands from east and north Japan. It also has tow warehouses at the plant and several stock points around Japan. Usually packages of canned cooking oil to the same region are batched and delivered to the stock point which covers the region. Then they are delivered to retail stores by using smaller trucks. However, demands from Kanto area, which is a region near one of the plant and with the largest demands, can also be covered by delivering packages directory from the plant. Thus the company has two alternative ways, direct/indirect delivery, to fulfill orders from a retail store in Kanto area. In Kanto area, the company has three stock points, covers about 20000 retail stores, and receives 300 to 500 orders every week day.

The company uses several outside distributors to deliver packages of canned cooking oil from warehouses and from stock points. The shipping charge is defined in the tariff offered by them, but the cost structure of the delivery from warehouse is quite different from that of the delivery from stock points. Since the delivery from a stock point is a rather short trip and a truck can make several trips a day, the shipping charge from a stock point is determined by the size of the order and the distance to the retail store. On the other hand, the delivery from a warehouse is usually a long trip and a truck can make at most one trip a day. Therefore, the shipping charge from a warehouse depends on the size of a truck and the distance to the retail store.

In this article we call a delivery to a retail store from a warehouse a direct delivery, and a delivery to a retail store from a warehouse via a stock point a indirect delivery. If a truck for a direct delivery is fully filled with orders, the unit transportation cost of a direct delivery is relatively smaller than that of a indirect delivery. Therefore we can reduce the total distribution cost by finding an appropriate group of orders packed into the same truck, and increasing the ratio of the direct delivery.

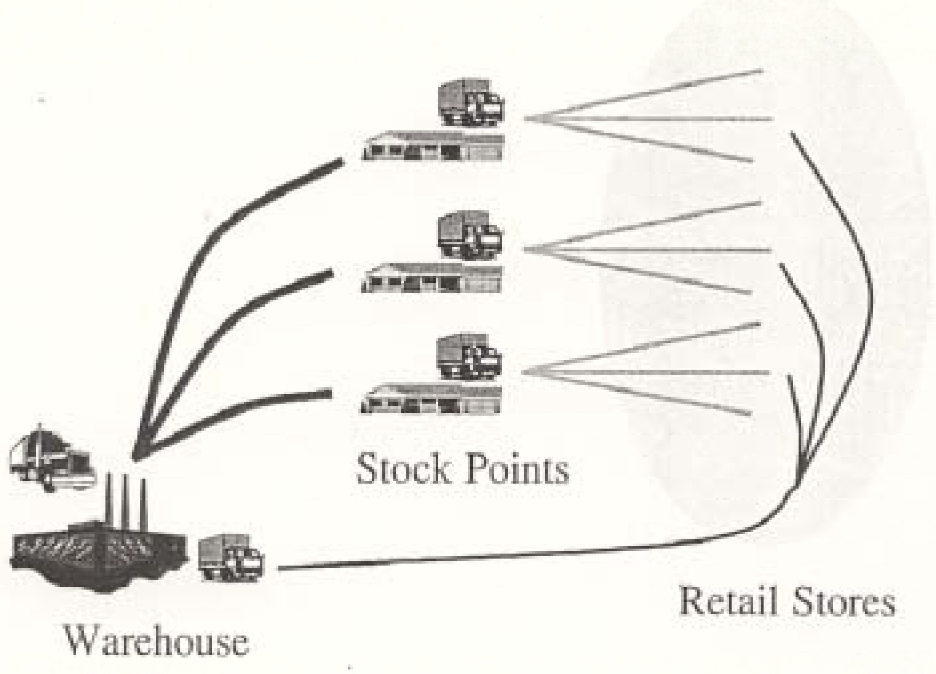


Figure 1: Direct/Indirect truck delivery

2 Modeling by bin-packing problem

2.1 Cost structure

The transportation cost from a warehouse and from a stock point have quite a different cost structure. We explain how the shipping charge is defined in the both cases.

The unit transportation cost from a warehouse is defined in terms of the distance to the destination and the size of a truck. Moreover minimum load of a truck is defined as 93.33% of the capacity of a truck. Even if the total amount of orders does not reach 93.33% of the capacity of a truck, we need to ensure the shipping charge for the minimum load. Thus the actual unit transportation cost from a warehouse can be reduced by appropriately grouping orders and filling trucks at least up to 93.33% of the capacity. Since a delivery from a warehouse is usually a very long trip, there are several constrains on a delivery from a warehouse:

- a truck must return to a warehouse within 16 hours including rest time and unloading time.

- a truck cannot make more than three stops in a trip.
- additional charge per stop is required after the third stop in a trip.
- transportation time must be less than 1 hour between stops.
- a truck must arrive at a stop between allowable time-window.

On the other hand, the unit transportation cost from a stock point is defined in terms of the distance to the destination and the size of an order. Since there is no additional condition on the shipping charge from a stock point, the transportation cost of an order is completely determined by the above two factors.

Thus the distribution cost of a direct delivery consists of only the transportation cost from a warehouse to a retail store, and the distribution cost of a indirect delivery consists of the transportation cost from a warehouse to a stock point, the handling cost at a stock point, and the transportation costs from a stock point to a retail store.

Distributors from a warehouse uses two sizes of trucks (4 ton and 10 ton). The unit shipping charge of the smaller truck is higher than that of the larger truck. The same distributors are used for the transportation from a warehouse to a stock point in the case of an indirect delivery. However, for the simplicity, we assume that the total amount of orders for an indirect delivery is much larger than the capacity of the larger truck and that all the orders are delivered in a truck of full load. Therefore, we can estimate the transportation cost from a warehouse to a stock point without knowing how they are packed together into a truck.

Once we accept the assumption above, the problem became more tractable. At first we treat all the orders as if they are delivered by the direct delivery, and find a packing of the orders into trucks under the above constraints. Then for each truck we can estimate the distribution cost for the both direct and indirect deliveries. If the distribution cost for the indirect delivery is cheaper, we release the orders packed into the truck. Thus we can decide the delivery method after assigning all the orders to trucks.

2.2 Multi-size bin-packing

We modeled the problem as an extension of a bin packing problem [1]. The size of an order corresponds to the size of an item, and the capacity of a truck to the size of a bin. In this problem, we deal with different sizes of trucks. A bin-packing problem with different sizes of bins is a multi-size bin-packing problem. Although there are other constrains we can expect heuristics for pure bin-packing problem will also works effectively to this problem.

We used two well known online packing procedures for pin-packing problem: First Fit (Figure 2) and Best Fit (Figure 3). These packing procedures check not only the capacity of a truck but the other constraints. When packing orders we assume the capacity of a truck is that of the largest truck. And when estimating transportation cost we assume a truck is the smallest one that can accommodate all the assigned orders.

Given an order, First Fit checks each truck in turn. If a truck is fond which can accept the order with all the constraints satisfied, First Fit packs it into the truck. Otherwise, First Fit uses a new empty truck. On the other hand, Best Fit checks all the trucks and evaluate the fitnesses of trucks. IF there is a truck which can accept the order with all the constraints satisfied, Best Fit packs it into the fittest truck. Otherwise Best Fit uses a new empty truck. In a pure bin-packing problem, Best Fit evaluate the fitness of a bin in terms of the remaining space of the bin. A bin with the least remaining space is estimated the fittest bin. Since, in our application, we are going to optimize the total distribution cost, we used the following two other fitness evaluations:

- Increase of the transportation cost
- Increase of the amount of goods delivered with unit cost

For the first fitness measure, a truck of the least increase is the best truck. For the other fitness measure, a truck of the most increase is the best truck.

First Fit packing procedure

```
FirstFit(O, T)
/* O: sequence of orders */
/* T: sequence of assigned trucks */
var assigned; /* flag if an order is assigned or not */
  foreach i ∈ O do
    assigned ← FALSE;
    foreach J ∈ T do
      if assigned = FALSE ∧ CheckConstraint(J, i)
        then J ← J ∪ {i};
             assigned ← TRUE fi;
    od;
    if assigned = FALSE
      then T ← T ∪ {{i}} fi;
  od;
```

Figure 2: First Fit packing procedure

3 Heuristic search methods

Heuristic methods are used to solve the problem. We start with a initial feasible solution, and iteratively improve the solution toward the global optimum. For improving a intermediate solution, we implemented three different heuristic search methods;

- Local search
- Sequencing genetic algorithm
- Grouping genetic algorithm

and evaluated their performance for our problem. In each iteration of the improvement, the best solution is registered.

3.1 Local search

Local search maintains a single instance of feasible solution and modify the instance randomly until an improved feasible solution is obtained. A initial feasible solution is constructed by applying a packing procedure to a sequence of orders in decreasing order of the size. In the

Best Fit packing procedure

```
BestFit(O, T)
/* O: sequence of orders */
/* T: sequence of assigned trucks */
var Trucks, /* a set of trucks */
    best; /* the fittest truck */
  foreach i ∈ O do
    Trucks ← ∅;
    foreach J ∈ T do
      if CheckConstraint(J, i)
        then Trucks ← Trucks ∪ {i} fi;
    od;
    if Trucks = ∅
      then T ← T ∪ {{i}};
       else best ← BestTruck(Trucks);
            T ← T ∪ {best} fi;
  od;
```

Figure 3: Best Fit procedure

pure bin packing problem, this heuristics gives a very good approximation of the optimal solution with asymptotic worst case ratio of 9/11.

In the modification phase, we randomly select trucks from the feasible solution with a certain probability P_t. Then, the orders in the selected trucks are released and packed into trucks again by applying a packing procedure.

The modified solution does not always a better solution. We iterate the modification phase until a better solution is obtained. If we fail to obtain a better solution more than predetermined number of times, we terminate the iteration.

3.2 Sequencing genetic algorithm

Sequencing genetic algorithm [4] treats a sequence of orders as a chromosome. A corresponding solution is obtained by applying a packing procedure to the sequence. The genetic operators modify the order of the sequence. In the pure bin packing problem, it is ensured that if packing procedure is fixed there exists a sequence that gives the optimal solution. However, we are not sure it is also true for our problem. The followings are the detailed descriptions of genetic operators.

Initial Population: To avoid homogeneous population, we generated the initial population by swapping the position of randomly selected pairs in the decreasing sequence of orders. The probability of the selection was set to 10%.

Selection: Tournament selection was used. A pair of sequences are randomly selected from the population, and their fitness values are evaluated by applying a packing procedure. The better one survives to the next generation.

Crossover: A pair of sequence are randomly selected from the current population as parents with a certain probability P_{SGc}. The two offspring are generated from them by a two point crossover and replaced with the parents. To ensure the uniqueness of entities in the sequence, we used cycle crossover (CX) [3] that preserves the position of orders in the parent sequence. CX was originally applied to the sequence of cities in the traveling salesman problem.

Mutation: An order in a sequence is randomly selected with a certain provability P_{SGm} and its position is swapped with another randomly selected order.

3.3 Grouping genetic algorithm

Grouping genetic algorithm [2] treats a feasible solution as a chromosome. Therefore a solution itself is directly modified by the genetic operations. This was shown to be very effective heuristics for a large bin packing problem in [2]. Since the grouping genetic algorithm is robust for any grouping problem, we can expect this heuristics preserves good groups of orders through generations even for our problem. The followings are the detailed description of the genetic operators.

Initial Population: To avoid homogeneous population, we generate the initial population by applying a packing procedure to a sequence of orders slightly modified from a decreasing one. The modified sequence is obtained by swapping the position of randomly selected pair of orders in the decreasing sequence. The provability of the selection was set to 10%.

Selection: Tournament selection was used. A pair of solutions are randomly selected from the population, and their fitness values are compared. The better one survives to the next generation.

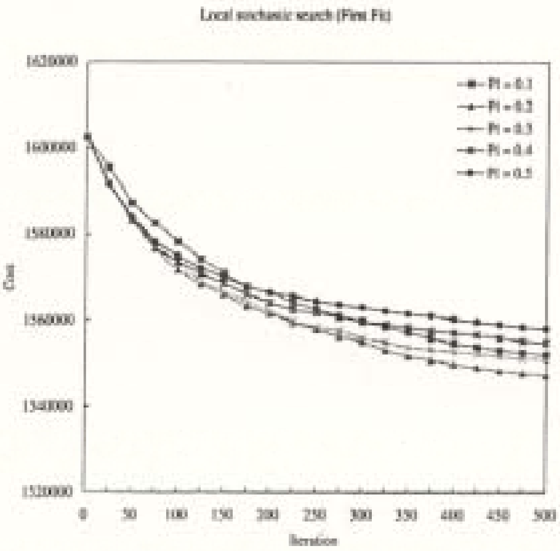
Crossover: A pair of solutions are randomly selected from the current population as parents with a certain probability P_{GGc}. Two offspring are generated from them by a tow point crossover and replaced with the parents. A solution is regarded as a sequence of trucks filled with orders. Parent sequences are divided into three partitions. An offspring is generated by merging the whole sequence of one parent and the center partition of the other parent. If there is a truck the order in which also appears in another truck, all the orders in such trucks are released and are again packed into trucks by a packing procedure.

Mutation: A truck in a solution is randomly selected with a certain probability P_{GGm} and all the orders assigned to the truck are released. Then the released orders are again packed into trucks by a packing procedure.

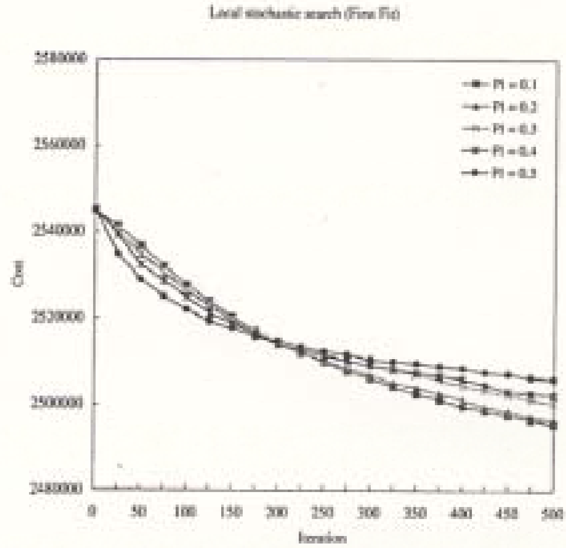
4 Experimental results

We applied the above heuristic search methods to the actual shipping orders of the company. We selected two days of the different characters for the experiment. One day has an average amount of orders. The other day has the largest amount of orders.

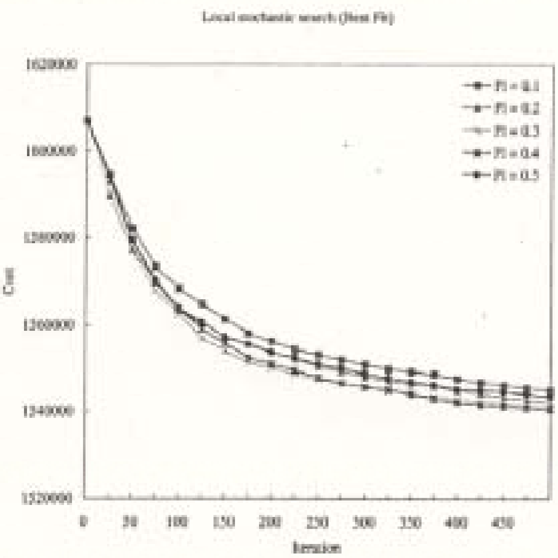
Figure 4 shows improvement processes of a solution by the local search for various values of P_l. Plotted values are averaged over 100 trials with different seeds for random number generator. Considering the performance of packing procedures, Best Fit outperformed First Fit in the both cases in terms of the speed of the improvement. Best Fit is also less sensitive



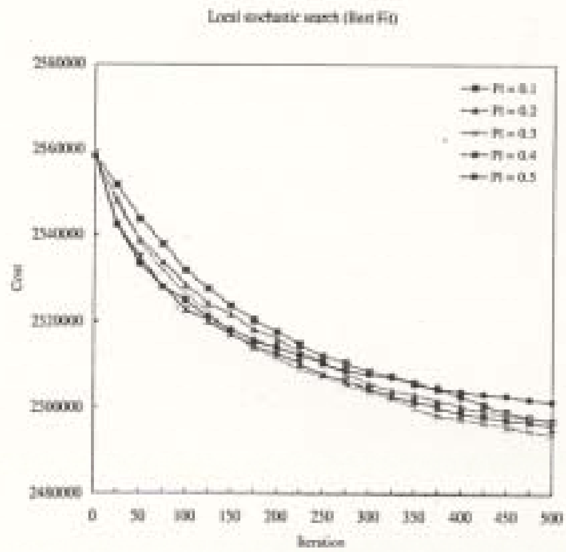
(a) First Fit (medium)



(b) First Fit (large)



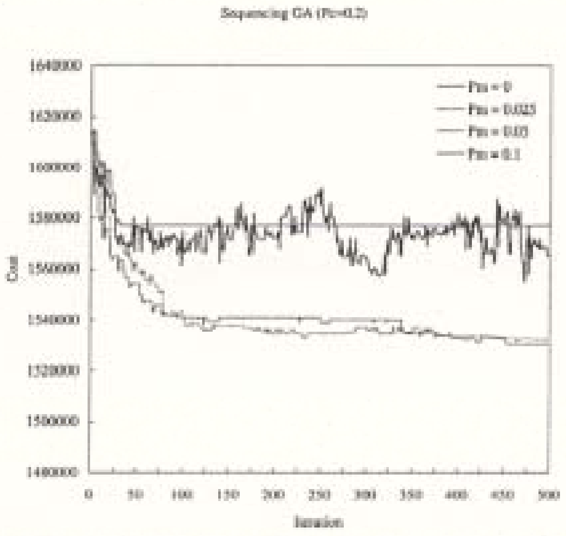
(c) Best Fit (medium)



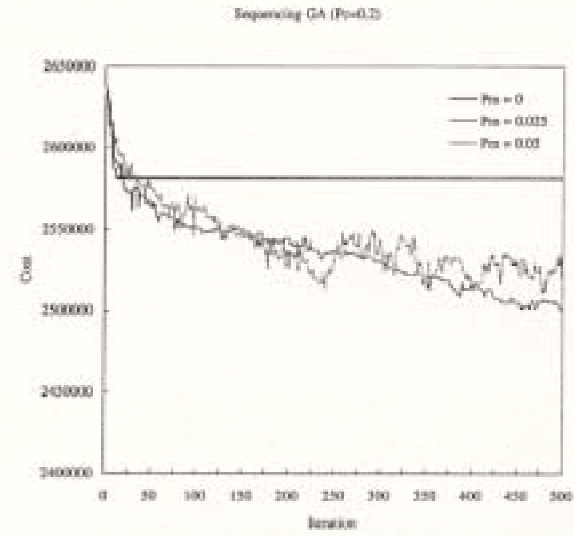
(d) Best Fit (large)

Figure 4: Local search

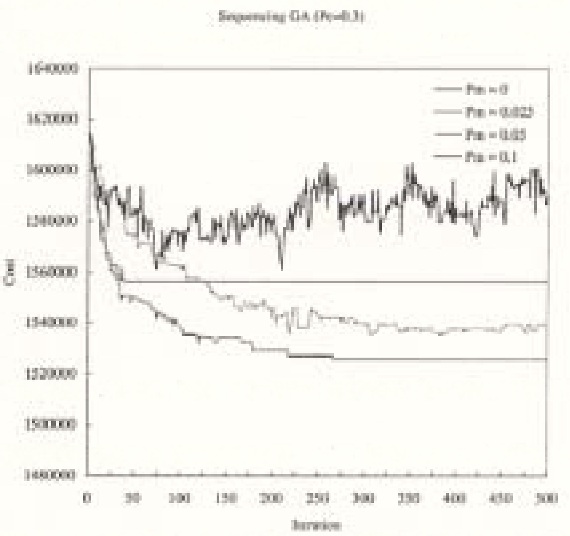
to the value of P_l. Since Best Fit assigns an order to the best truck among allowable ones, the solution does not seem to depend so much on how many orders are released and assigned again. On the other hand, Fist Fist is sensitive to the value of P_l. When P_l is small, the speed of the improvement is slower because only small part of the solution is changed in the modification phase. However, large values of P_l do not show good performance, neither. Release of too many orders likely results in a similar solution. As a result, $0.2 \sim 0.3$ is appropriate for the value of P_l in these cases.



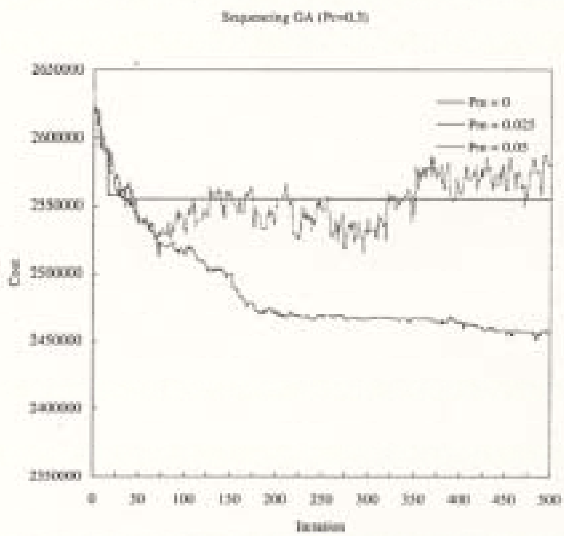
(a) $P_{SGc} = 0.2$ (medium)



(b) $P_{SGc} = 0.2$ (large)



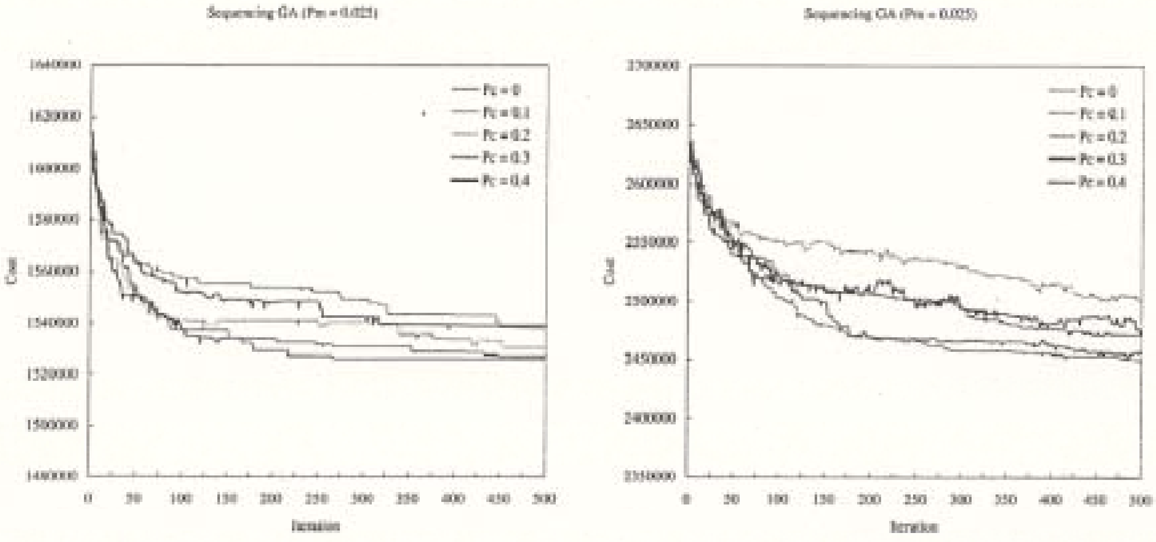
(c) $P_{SGc} = 0.3$ (medium)



(d) $P_{SGc} = 0.3$ (large)

Figure 5: Sequencing genetic algorithm for fixed crossover rates

Figures 5 and 6 show improvement processes of a solution by the sequencing genetic algorithm for various values of crossover rate P_{SGc} and mutation rate P_{SGm}. The size of the population was fixed to 100 for all the cases. The plotted values are the best solution in the



(a) $P_{SGm} = 0.025$ (medium) (b) $P_{SGm} = 0.025$ (large)

Figure 6: Sequencing genetic algorithm for fixed mutation rates

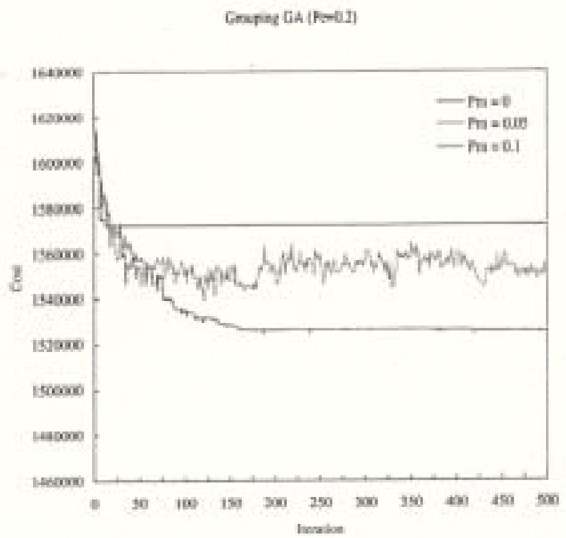
population. Figure 5 shows improvement processes when the crossover rate is fixed to 20% and 30%. From the figure we can observe that the sequencing genetic algorithm is very sensitive to the mutation rate P_{SGm}. The sequencing genetic algorithm without the mutation rapidly reaches the premature convergence, and is almost useless. Also high mutation rate brings an almost random behavior to the algorithm. Thus the mutation rate P_{SGm} of around 2.5% is appropriate for our problem. Figure 6 shows improvement processes when the mutation rate is fixed to 2.5%.

Figures 7 and 8 show improvement processes of a solution by the grouping genetic algorithm for various values of crossover rate P_{GGc} and mutation rate P_{GGm}. The size of the population was fixed to 100 for all the cases. The plotted values are the best solution in the population. Figure 7 shows improvement processes when the crossover rate is fixed to 20% and 30%. From the figure we can draw the same observation as the sequencing genetic algorithm, that is, the grouping genetic algorithm is also sensitive to the mutation rate. However the grouping genetic algorithm is insensitive to the crossover rate as shown in Figure 8 that shows improvement processes when the mutation rate is fixed to 5%. In other words, the crossover operation does not contribute to the solution improvement in the grouping genetic algorithm.

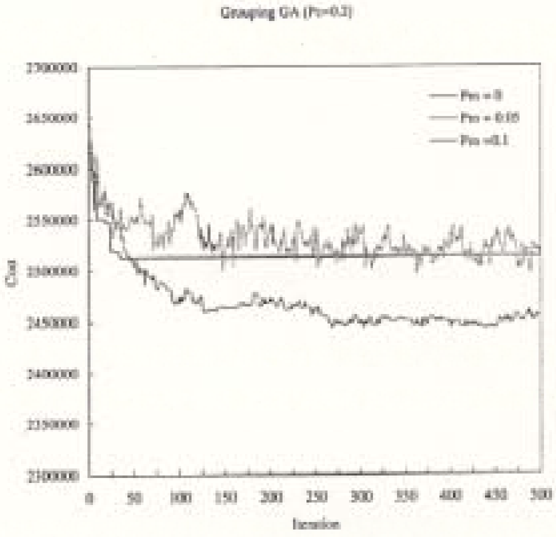
5 Discussion

We applied three different heuristics to our optimization problem. Whenever we adopt heuristic approach to the combinatorial optimization problem, we are interested in the optimality of the obtained solution. In our problem, theoretical bounds are hard to be obtained because of complicated constraints. We estimated a lower bound of the optimal value for instances of the problem, and compared to the solution obtained by the heuristic methods.

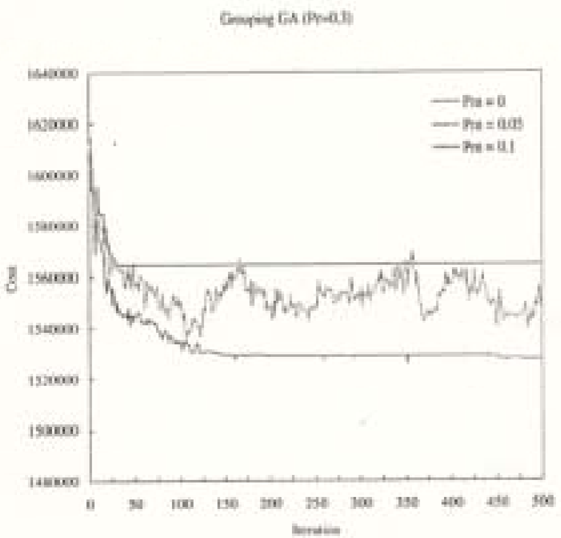
To obtain the lower bound of the optimal solution, we formulated the problem as an integer programming (IP) problem and solved it by a linear programming (LP) relaxation. By



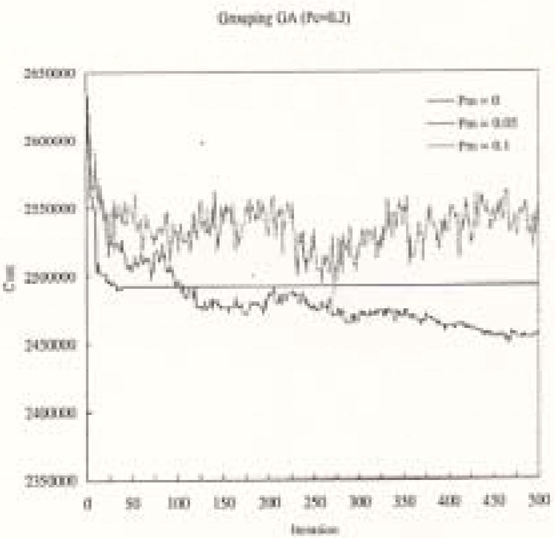
(a) $P_{GGc} = 0.2$ (medium)



(b) $P_{GGc} = 0.2$ (large)

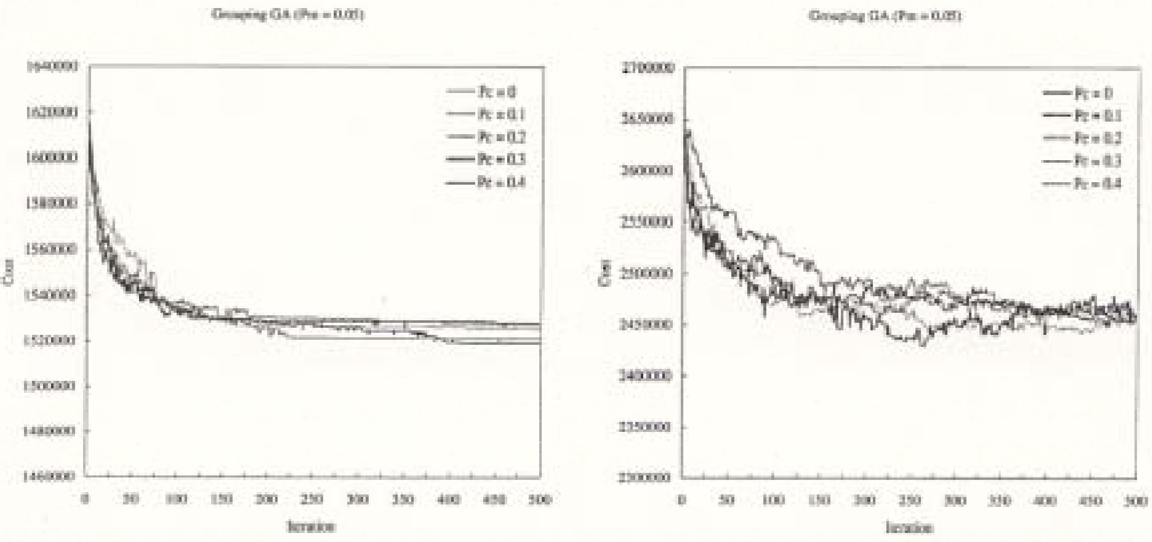


(c) $P_{GGc} = 0.3$ (medium)



(d) $P_{GGc} = 0.3$ (large)

Figure 7: Grouping genetic algorithm for fixed crossover rates



(a) $P_{GGm} = 0.05$ (medium) (b) $P_{GGm} = 0.05$ (large)

Figure 8: Grouping genetic algorithm for fixed mutation rates

enumerating all the feasible set of orders that can be assigned to a truck, we can formulate the problem as a typical set partitioning problem. The number of the feasible set of orders is considerably large in a usual truck scheduling problem. However, in our case, the number of stops allowed to a truck is constrained. At the same time, we aggregated orders to the same retail store and further reduced the number of order combinations. Thus the number of the feasible set became a tractable size. IP formulation is as follows.

$$\text{IP:} \qquad \text{minimize} \qquad \sum_{j=1}^{n} c_j x_j, \tag{1}$$

$$\text{subject to} \qquad \sum_{j=1}^{n} a_{ij} x_j = 1, \qquad i = 1, \ldots, m, \tag{2}$$

$$x_j \in \{0, 1\}, \qquad j = 1, \ldots, n, \tag{3}$$

- m = the number of feasible sets of orders that can be assigned to a truck
- n = the number of orders
- c_j = transportation cost for a feasible set of orders j.
- $x_j = 1$ if a feasible set of orders j is assigned to a truck; 0 otherwise.
- $a_{ij} = 1$ if an order i is covered by a feasible set of orders j; 0 otherwise.

Transportation cost c_j is the minimum value among direct and indirect deliveries.

We can obtain LP formulation by relaxing the integral condition of the decision variables

x_j $(j = 0, \ldots, m)$ as follows.

$$\text{LP:} \quad \text{minimize} \quad \sum_{j=1}^{n} c_j x_j, \tag{4}$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j = 1, \quad i = 1, \ldots, m, \tag{5}$$

$$0 \le x_j \le 1, \quad j = 1, \ldots, n, \tag{6}$$

Table 1 shows lower bounds and solutions obtained by Grouping GA. The number of rows means that of orders. Because orders to the same retail store is aggregated together, the number of orders in the table is about a half of the actual number. The number of columns means that of feasible set of orders assigned to a truck.

Data	#Row	#Column	Lower Bound	Grouping GA	Error (%)
VP0401	150	10892	1,499,876	1,515,246	1.02
VP0402	155	15152	1,281,804	1,317,117	2.75
VP0403	147	5661	1,144,521	1,148,329	0.33
VP0404	201	16613	1,963,540	1,999,458	1.83
VP1203	256	60192	2,521,645	2,610,851	3.54
VP1204	291	53496	2,230,362	2,260,486	1.35
VP1205	222	26547	1,649,077	1,698,057	2.97
VP1206	319	76362	2,392,769	2,429,306	1.53

Table 1: Comparison to lower bounds

The performance of Grouping GA depends on the problems. In the problem, two different sizes of trucks are used, and we determine the size of truck depending on the amount of orders assigned to the truck. Thus, the packing procedure cannot judge which is the fittest truck appropriately.

6 Summary

We described heuristic methods for optimizing the distribution cost of truck deliveries. According to the results of the experiments on the company's historical data, Grouping GA is the most effective heuristics among the three methods. Sequencing GA is effective for a pure bin-packing problem, but showed poor performance for this sort of problem with a lot of constraints. We estimated the cost reduction by applying our methods to the actual orders of the company. On average, 14% cost reduction was achieved.

The company assigned orders larger than 70 cases to the direct delivery. However, from the detailed analysis, we found that most of orders larger than 40 cases are assigned to the direct delivery in the simulation. Also the ratio of the direct delivery increased from 60% to 80%.

References

[1] Jr. E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorihms for bin packing: A survey. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 2, pages 46–93. PWS Publishing Compary, 1997.

[2] Emanuel Falkenauer. *Genetic Algorithm and Grouping Problems*. John Wiley & Sons, 1998.

[3] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.

[4] Colin Reeves. Hybrid genetic algorithms for bin-packing and related problems. *Annals of OR*, 63:371–396, 1996.