Synchronizing inventory and transport within supply chain management

Report on the problem presented by the

Infora Research Group

on the

99th European Study Group with Industry, Novi Sad, February 2014

Tivadar Danka, Attila Dénes, Géza Makay, Kyeongah Nah, Zsolt Vizi

Bolyai Institute, University of Szeged, Hungary

1 Introduction

We study a problem presented by the Infora Research Group about distribution of tobacco in Serbia. The problem is to find the most cost-efficient way to distribute products from the production center in Senta to the customers. After leaving the production center, products can be stored in warehouses before being redistributed to cross-docking points. Cross-docking points are used to change the means of delivery (e.g. truck to van), but they cannot store large amount of goods. The following is the description of factors related to the supply costs.

1. *Locations of facilities*: There is one production center in Senta. Currently, there are 5 warehouses and 16 cross-docking points (see Figure 1), which can be relocated for optimality. There are 16141 customers.

2. *Renting cost for facilities*: Monthly renting cost for warehouses in Belgrade, Niš and Novi Sad is 3.5 €/m^2, and 2.75 €/m^2 for warehouses in other cities. Renting cross-docking points costs 100 € for a month.

3. *Capacity of warehouses*: Currently the capacity of the warehouse is 1500 m^2 in Belgrade, and 1000 m^2 in Niš, Novi Sad, Kragujevac, and Požarevac.

4. *Means of transportation*: The delivery from the production center to the distribution points is outsourced. Trucks are used from the distribution centers to the cross-docking points, and vans are used for the delivery to the costumers. There are two types of trucks with 3.5t and 5t capacity, and one type of van with 1.5t capacity.

5. *Costs for transportation*: From a warehouse to a cross-docking point and a cross-docking point to customers, costs should be paid for round trips. From the production center to the warehouses, cost is paid for one-way only. The cost is 35 ¢/km for 5t trucks, 30 ¢/km for 3.5t trucks and 20 ¢/km for 1.5t vans.

6. *Customer needs*: One van with 70–80% load can serve about 100 customers in a city in an eight-hour shift. All customers get the same amount of goods for each delivery, and the actual amount of delivered goods is determined by the frequency a customer is visited with.

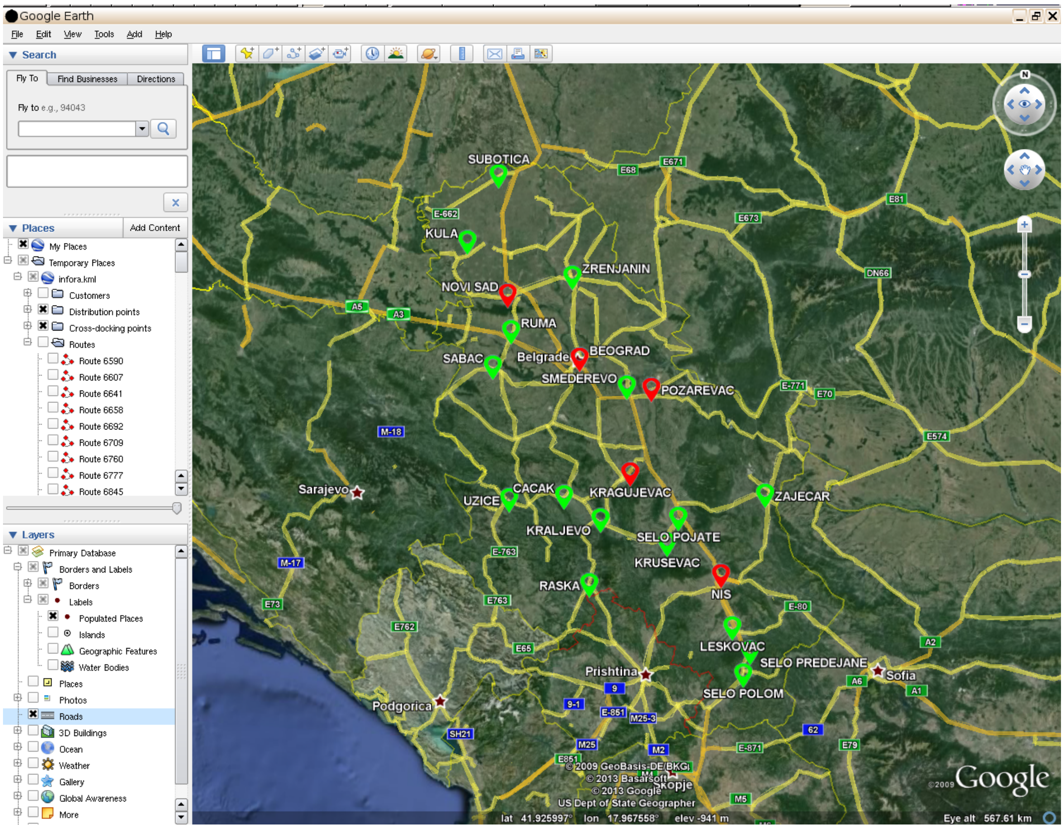

Figure 1: Current location of warehouses and cross-docking points.

Based on the factors listed above, we optimize the supply link, such as the number of warehouses/cross-docking points, their locations, and the optimal route of the delivery.

We have received a table from the Infora Research Group containing the data of all routes: the shops' addresses with geographical coordinates, the warehouses they are served from and the route(s) on which they are served. We converted the table to a database, and created a Google Earth ([1]) presentation from the given data. We found some problems:

- missing coordinates for some of the customers,
- some of the coordinates are incorrect (e.g. a customer in Bulgaria),
- some of the customers are assigned to routes which are not suitable for them as they lie far from other customers on the same route (see Figure 2),
- sometimes the order of the customers visited on the same route is illogical, as part of the route consist of driving back and forth between two regions instead of visiting customers within one region and then moving to the next region (see Figure 3),
- the database is incomplete in the sense that the cross-docking points are not included in the routes and the corresponding cross-docking points are not assigned to the routes; routes between the production place and the warehouses, resp. between the warehouses and the cross-docking points are not given.

We give an overview of a genetic algorithm in section 2. In section 3 and 4, genes and mutation for our problem is described in detail. The advantages and disadvantages of using genetic algorithm is addressed in section 5. In section 6, we present ideas and improvements for the implementation.

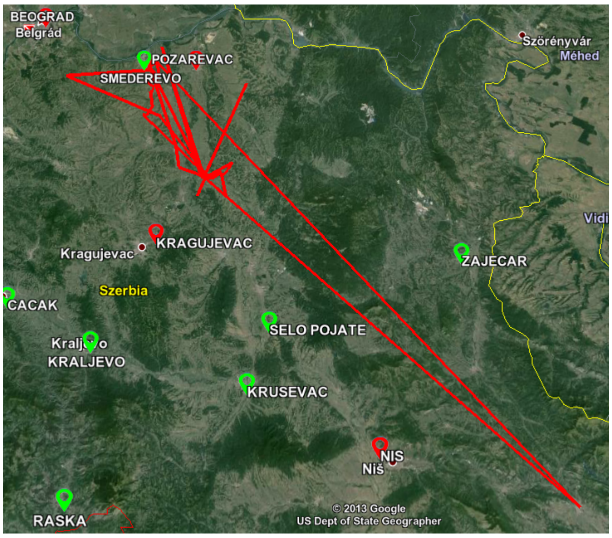

Figure 2: Incorrect route assignment: one customer is too far from the rest

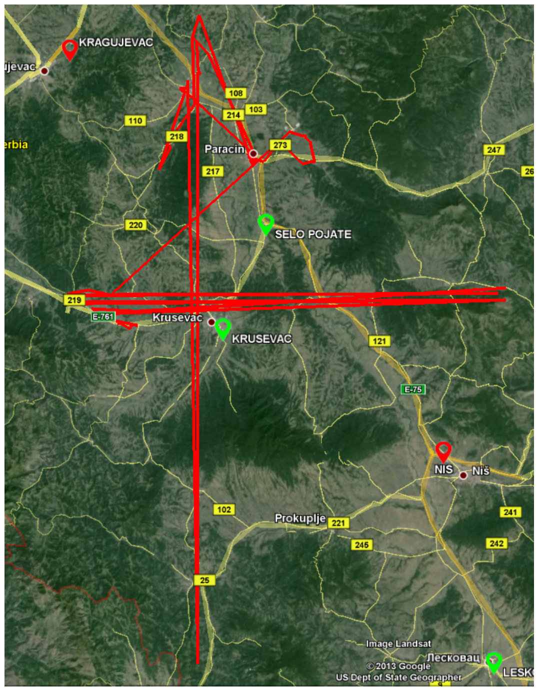

Figure 3: Incorrect route assignment: going back and forth between two regions

2 Overview of the genetic algorithm

The genetic algorithm ([2]) is a nondeterministic algorithm designed to find optimal or near-optimal solutions for optimalization problems by imitating the evolution. For an optimization problem, we can think of the possible solutions as a species, the variables of the problem as a predetermined set of genes, and the cost function as a fitness score of the individual solutions. Given a set of possible solutions, the algorithm executes three consecutive steps: crossing of the solutions, mutation and selection.

2.1 Crossing

When two individuals reproduce, they share their genes, therefore providing the opportunity for the offspring to be better than both of its parents. During the crossing stage, we create offsprings by randomly mixing the genes of the selected individuals, and form the next generation from them. Without crossing, the average fitness score of the population would increase very slowly.

2.2 Mutation

Mutation is basically changing each gene of an individual with a small probability. During reproduction, this gives an opportunity to increase the genetic variation of the population. Without mutation, the set of possible genes would be very small, since crossing introduces no variation, only a mixture of genotypes.

2.3 Selection

In each generation, we shall select the fittest individuals to reproduce, and form the next generation from their offsprings. Without selection, it can happen that "bad genes" persist for a long time, therefore disrupting the convergence to the optimal solution. After the new generation is created, the old one is discarded, and the process is repeated for a predetermined number of generations, when an appropriate fitness is reached, or until the fitness score does not change significantly for a few generations. The last generation of the solutions shall be near-optimal.

3 Genes for this problem

3.1 Genes for warehouses (WH) and cross-docking points (CDP)

There are 39 cities in Serbia with population higher than 20000. We assume that cross-docking points and warehouses are located in these cities – if needed, we can add any other place, or even exact locations. The gene containing the information about the location of WHs and CDPs will be an array of 39 integers. The possible states are:

- 0: no WH or CDP in the city,
- 1: WH,
- 2: CDP.

Along the genes, we need to store additional information about costs. The cost of a WH depends on its square footage, which will be computed from the demand, but the CDPs have fixed costs.

3.2 Supplying routes from the production center to warehouses

There are two pieces of information we have to store:

- the frequency of visits: a bitvector of seven bits, one for each weekday,
- the WHs visited: a bitvector of 39 bits.

The order in which the WHs are visited are determined by solving a one way travelling salesman problem.

3.3 Supplying routes from warehouses to cross-docking points

Every CDP must be supplied from a warehouse with a 3.5t or a 5t truck, and multiple CDPs can be supplied in one route. We need to store the following information for each route:

- Starting warehouse
- The frequency of visits: a bitvector of 7 bits, one for each weekday
- The type of truck used: with the 5t or 3.5t loading capacity
- The CDPs visited: a vector of 39 bits

The order in which the CDPs are visited are determined by solving a travelling salesman problem (TSP, [3]).

3.4 Supplying routes from cross-docking points to customers (CU)

There are 16141 customers which must be supplied with given frequency. Every customer is supplied by a van. For each supplying route, we store the following information:

- The frequency of visits: a bitvector of 7 bits, one for each weekday
- The CUs visited: a bitvector of 16141 bits

The order of the visited customers is once again determined by solving a TSP.

3.5 Conditions on the genes

For a genotype to provide a valid solution, some conditions have to be met.

- All WHs and CDPs must be visited. Although this condition is not really necessary since an unvisited CDP is extra cost and therefore it will be eliminated eventually by the genetic algorithm, enforcing this condition makes the algorithm more efficient.
- Every CU must be visited with at least the frequency they require.
- The demands of the CDPs are computed from the routes and the number of served customers. These demands must be fulfilled by the routes from WHs to CDPs and the demands of the WHs can be calculated from this. These demands must be fulfilled by the routes from the production center to WHs.

4 Mutations and crossovers

4.1 Mutations

Mutations in the genes storing warehouse and cross-docking point locations can be simply changing the status of every city with some probability. This means that in each city, we can

- set up a new warehouse or cross-docking point,
- promote cross-docking points to warehouses,
- demote warehouse to cross-docking point,
- eliminate an existing warehouse or cross-docking point.

The mutations on the routes are more than flipping bits. The possible mutations are

- change frequency of the route,
- add or remove waypoints on the route,
- change the truck used on the route from a WH to CDPs,
- add a new route,
- remove a route.

4.2 Crossovers

For genes containing the location of warehouses and cross-docking stations, we can do the crossover by slicing up the two genes to parts of the same size, and then swap them. This means that for the offsprings, we choose some warehouses and cross-docking points from the mother, and some others from the father. This method also works for genes containing information about the supplying routes: we choose some routes from the mother, and we choose some from the father. In this case, we also have to make sure that the offspring genes satisfy our conditions regarding the routes.

5 The advantages and disadvantages of the genetic algorithm

5.1 Advantages

(i) The genetic algorithm can solve complex optimization problems, such as this one. We shall optimize the location of the cross-docking points and the supplying routes the same time, because optimizing one and then the other may result in highly sub-optimal solutions.

(ii) It can handle problems with a large number of variables, and it does not need to know the connection between them.

(iii) Due to the selection process and the large variation of genotypes granted by mutation, the convergence of the algorithm is fast.

(iv) It can be easily implemented to run in parallel using multicore processors, GPUs, application accelerators or multiple computers.

5.2 Disadvantages

(i) The algorithm does not find an exact solution, but instead a bundle of near-optimal ones.

(ii) It may also find sub-optimal solutions, because it can converge to local extrema. A solution to this problem is to run the algorithm multiple times, starting from a different set of initial solutions.

(iii) For this particular problem, there are many genes and it may prove difficult to code them efficiently. For example, it is possible that after crossing and mutating genes, we do not obtain a valid solution (some customers may be left unsupplied, we can have unused cross-docking points, etc). To avoid this, we have to check the validity of every genotype obtained by crossing and mutating existing solutions.

6 Ideas for implementation

To decrease the number of genes in our problem it is feasible to handle small towns with few (e.g. at most 25) customers in one route, as in a route connecting smaller towns, moving between the customers inside a town is negligible in comparison with the distance between the towns. 92% of the towns fall in this category, which means a significant decrease in the number of customers to be handled separately.

It seems logical to handle customers in cities where a warehouse or a cross-docking point is located, from that warehouse or cross-docking point. This means solving a multipath travelling salesman problem in the candidate cities before we start the genetic algorithm. This method decreases the number of customers to be handled individually even more: after applying this and the previous idea together only 17% of the customers remain.

We can also decrease the number of genes by assuming that all customers are covered only from a cross-docking point which is at most twice as far as the nearest cross-docking point, and all cross-docking points are covered from a warehouse which is at most twice as far as the nearest warehouse.

We can also fix the location of the warehouses (if moving them is too costly) and the cross-docking points, and even (part of) the routes if necessary, the genetic algorithm easily allows that.

For speeding up the computation, we can cache the shortest paths and TSP results which have already been computed as well as keep track of the costs and recompute them only if they are changing.

The algorithm does not change if we change anything here, e.g.: the frequency from weekly to biweekly or monthly, the candidate cities for warehouses and cross-docking points, the number of customers, the actual costs, the number and type of trucks and vans, etc. The genetic algorithm can also handle more complex conditions if necessary, e.g. delivery times, actual orders of the customers, etc.

7 Conclusions

The problem presented by the Infora Research Group is quite complicated and challenging. It contains location and route optimization tasks within a supply chain to achieve the lowest cost possible. In order to start with the implementation we would need the correct actual routes, so that we know the exact problem, and we can compare our results with the current costs. We hope to cope with this problem using the genetic algorithm, which seems suitable for handling the several variables used here.

8 Acknowledgements

We would like to thank the companies for the interesting problems, especially the Infora Research Group and its representative, Nebojša Gvozdenović, who was very supportive during the conference. This was our first, but certainly not the last participation on the conference series *European Study Group with Industry*, we enjoyed it very much. We would also like to thank the members of the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, for organizing this conference.

References

[1] Google Earth: http://www.google.com/earth/

[2] Genetic algorithm: http://en.wikipedia.org/wiki/Genetic_algorithm

[3] Travelling salesman problem: http://en.wikipedia.org/wiki/Travelling_salesman_problem