Optimization of a Supply Chain Distribution Network using MIP

Ramiz Assaf, Andrea Ratti and Anteneh Teferi

June 11, 2010

Mechanical Engg. Dept.
Politecnico di Milano
via la Masa 1 , 20155 Milano
Tel.:0039-22399-8534
ramiz.assaf (at) mail.polimi.it

Introduction

In last years, *quality* and *logistics* issues have represented real challenges for competitiveness and survival of modern companies, that renewed the attention on the design and management of production systems. While production problems, i.e. production scheduling and procurement, have been fundamentally solved during last years, today's challenge is focused on goods mobility (or *logistics*), in terms of dispatch the right amount of products to right customer in the fastest way possible. The objective of this work is to study, comprehend and model a supply chain distribution network, and to formulate a proper problem statement which highlight every feature of the real system. A step by step approach is used in order to describe the problem from a basic ideal case to a more realistic one, by reducing assumptions and widening of modeling capabilities. Different approaches are used to evaluate the formalized problem and finally comparison, evaluation and conclusion on the approaches used are presented. It is very important to point out that this work includes features concerning logistics topic (e.g. transportation costs, stocks, transportation vehicles, etc), considering pure production issues with a simpler representation. In this way, typical production schedule problems (e.g. scalable capacity, turnover and shifts, outsourcing manpower, etc) are not considered.

Contents

1 Basic Supply Chain

1.1 Assumptions

We consider two-level supply chain, composed by F *factories* and S *Sales points*. For each factory $i \in F$ we consider a fixed capacity C_i, while each sales point $j \in S$ has a demand d_j that should be satisfied. Each factory node is connected to each sales point node by an arc a_{ij} with a cost e_{ij} for every product item sent. Given this topology, we have to minimize the total costs for satisfying the whole demand of each sales point with an amount q_{ij} sent from any factory i to each specific sales point j. The quantity sent from factory i must be lower or equal to the capacity of that factory C_i. The received quantity at sales point j must be equal to the demand d_j of the sales point j. The total capacity of factories must be greater or equal to the total demand at the sales point.

1.2 Formulation

From a mathematical point of view, the problem could be formulated as following:

$$\min \sum_{i=1}^{F}\sum_{j=1}^{S} q_{ij}e_{ij} \tag{1}$$

where the decisional variable is q_{ij} subject to:

$$\sum_{j=1}^{S} q_{ij} \leq C_i \qquad \forall i \tag{2}$$

$$\sum_{i=1}^{F} q_{ij} = d_j \qquad \forall j \tag{3}$$

$$\sum_{i=1}^{F} C_i \geq \sum_{j=1}^{S} d_j \tag{4}$$

$$e \in \mathbb{R}^+ \text{ and } d, C \in \mathbb{N} \tag{5}$$

Decision Variables:

$$q \in \mathbb{N} \tag{6}$$

2 Basic Supply Chain with Lost Sales

2.1 Assumptions

Given the assumptions proposed in Section 1.1, we now consider an added complexity that is the option of not satisfying the demand d_j at a certain sales point by a quantity ς_j, but with a cost penalty L_j that depends on the sales point j considered.

2.2 Formulation

From a mathematical point of view, the problem could be formulated as the one in Section 1.1 where Equation 1 becomes:

$$\min \sum_{i=1}^{F}\sum_{j=1}^{S} q_{ij}e_{ij} + \sum_{j=1}^{S} L_j\varsigma_j \tag{7}$$

where the decisional variable added for this formulation is ς_j and the added part of Equation 7 indicates the total amount of lost sales at sales point j. In the same way, Equation 3 should be modified to include lost sales amount at each sales point j as:

$$\sum_{i=1}^{F} q_{ij} + \varsigma_j = d_j \qquad \forall j \tag{8}$$

To grant lost sales addition, Equation 3 should be relaxed. Lost sale costs L and lost sale products ς are added to variable domain as:

$$L \in \mathbb{R}^+$$

Decision Variables:

$$\varsigma \in \mathbb{N}$$

3 Basic Supply Chain with Lost Sales and Single Modal Transportation Costs

3.1 Assumptions

In previous section we used a variable cost per unit of product shipped from a factory i to a sales point j, namely e_{ij}. In real supply chains this transportation activity is normally done with trucks or with other means of transport, that the total transportation costs will depend on the number of vehicles used to undertake transportation activity. With these assumptions, N_{ij} is the number of vehicles used to ship a certain amount of product q_{ij} from factory i to sales point j, given a nominal vehicle capacity φ and vehicle transportation costs k_{ij}. In addition, we're assuming that only one vehicle type can be used and its nominal capacity φ is fixed..

3.2 Formulation

From a mathematical point of view, the problem could be formulated as the one in Section 2.1 where Equation 7 becomes:

$$\min \quad \sum_{i=1}^{F} \sum_{j=1}^{S} N_{ij} k_{ij} + \sum_{j=1}^{S} L_j \varsigma_j \tag{9}$$

A new constraint is added to calculate the proper number of vehicles for each route from i to j as:

$$q_{ij} \leq N_{ij} \varphi \qquad \forall i, j \tag{10}$$

The number of vehicles N, transportation costs k and vehicle capacity φ are added to variable domain as:

$$k \in \mathbb{R}^+ \quad \varphi \in \mathbb{N}$$

The added decision variable is:

$$N \in \mathbb{N}$$

4 Master Distribution Schedule

4.1 Assumptions

To achieve a realistic representation of a distribution network from an aggregate point of view, a proper planning on temporal horizon should be considered. In this way, given basic foretasted data concerning sales point demand and factories production capability, is possible to formulate a proper *master distribution schedule* on a temporal horizon T (e.g. semester) with a time bucket detail t (e.g. month). From this new set of assumptions, the objective is now to minimize the total transportation costs under a temporal horizon T.
The main introduction along temporal horizon is represented by *stock quantity* at time t, named σ^t, for each factory i and for each sales point j. With this new decisional variable we are assuming that factories and sales points are capable to store some products, incurring an holding cost equal to h. It is very important to highlight how in this formulation the *stock out*[1] effect isn't explicitly considered, but still roughly included using lost sales like a proxy.
Concerning factories, we are introducing the possibility to decide to activate or not the production during a time unit t, under a setup cost ϱ_i. Concerning sales point, we are introducing the possibility that the stock σ held between two time period is used to satisfy the demand of the next period.
To avoid any misunderstandings, a proper description of time planing is carried out. Given a generic time unit t, demand data are known at the beginning of the time unit. All decisions should be taken at the beginning of the time unit, as well as the transportations activities during the time t, while only evaluation of stocks σ is done at the end of the time unit t.

4.2 Formulation

From a mathematical point of view, the problem could be formulated as the one in Section 3.1 where Equation 9 becomes:

$$min \sum_{t=1}^{T} \left(\sum_{i=1}^{F} \left(h_i^t \sigma_i^t + \varrho_i^t y_i^t \right) + \sum_{j=1}^{S} \left(h_j^t \sigma_j^t + L_j \varsigma_j \right) + \sum_{i=1}^{F} \sum_{j=1}^{S} N_{ij}^t k_{ij} \right) \tag{11}$$

Where the decisional variable added for this formulation are σ_i^t and y_i^t. In the same way, Equation 8 should be reformulated like the placement between the stocks at time t and $t-1$, the whole incoming products, the lost sales and the demand as:

$$\sigma_j^{t-1} + \sum_{i=1}^{F} q_{ij}^t + \varsigma_j^t - d_j^t = \sigma_j^t \quad \forall j, \forall t \tag{12}$$

In the same way, Equation 2 should be reformulated to include production activation and stock existing between time t and $t-1$ as:

[1]The *stock out* is known in logistics literature as that particular phenomenon in which a certain amount of demand isn't satisfied due to stock or delivery problem. This is considered an extremely disruptive event for the performance evaluation of logistics systems.

$$\sum_{j}^{S} q_{ij}^{t} + \sigma_i^t = C_i y_i^t + \sigma_i^{t-1} \quad \forall i, \forall t \tag{13}$$

It is important to highlight when a factory is activated, its output production will be at its maximum. So we need Equation 10 and to add new variables inside a proper domain as:

$$k, h, \varrho \in \mathbb{R}^+ \text{ and } \varphi, L \in \mathbb{N}$$

So the complete decision variables are:

$$\sigma, N, \varsigma, q \in \mathbb{N} \text{and } y \in [0, 1]$$

5 Application of MIP solving methods

After formulating the problem in different way, the research team thinks it is proper to use the formulation proposed in Section 4.1 and try to use MIP approaches to solve it, then comparing the performance of these approaches in terms of accuracy and computational effort required. Three methods are implemented within this work: Branch & Bound, Greedy algorithm and Greedy algorithm quicken with a Local Search. For each method a proper software implementation has been proposed and benchmarked after small but complete reference case; this enable the research team to check and debug the software implemented prior to do the real evaluation phase. Besides, a complete comparison between the selected methods has been carried out, to achieve a consistent and homogeneous evaluations.

5.1 Reference example

We consider a small example to test the formulation proposed. Lets assume that we have a supply chain network composed of four factories and five sales point, where our goal is to minimize the total costs of delivering the products within a horizon of three time periods, given demand and capacity of each sales point and factory. Besides, a vehicle capacity φ of seventy units is assumed. Further information concerning input data are reported in Tables 1,2 and 3.

Table 1: Information regarding the Sales Points

	Sp1	Sp2	Sp3	Sp4	Sp5
Demand/time					
$d_{t=1}$	35	35	60	80	100
$d_{t=2}$	25	100	125	55	20
$d_{t=3}$	150	120	100	60	110
Lost sale cost / Unit:	10	15	20	15	17
Storage cost / Unit	12	14	23	50	8

Table 2: Information regarding the Factories

	F1	F2	F3	F4
Capacities	300	500	250	150
Setup Cost	4500	2000	2500	3000
Holding cost / Unit	5	2	4	3

Table 3: Vehicle Transportation Cost from each Factory to each Sales Point

	Sp1	Sp2	Sp3	Sp4	Sp5
F1	100	400	800	1100	1700
F2	1500	1800	1200	2200	1500
F3	1400	1000	400	1000	700
F4	800	700	900	700	1200

5.2 Branch & Bound method

IBM ILOG OPL v5.3 optimization program was used to formulate and solve the given problem with Branch and Bound method. The final solution obtained was 26975 Euros. In Appendix A, a complete output of the ILOG program will be reported showing the values of each decision variable along the whole formulation of the problem.

5.2.1 Effect of Vehicles Capacity on the computing time

Here, the effect of increasing or decreasing the number of linked arcs is investigated. Changing the vehicle capacity constant φ will have a direct effect on the links; e.g. if φ is equal to 1000, we are sure (for our example) that N_{ij} is either 0 or 1, but if the value of φ is 10 we are almost sure that the N vehicles are more than 1 vehicle and this could lead to products coming from different destinations which probably will force the solution to take longer than the previous case. But if the second case (low vehicle capacity case) is very small could be very fast as it is very similar to the relaxed problem, and the solution could be found also in a fast way.

To test these two effects, we consider a bigger case (Planning Horizion=8) then we started to change the vehicle capacity φ and accordingly changing the transportation cost matrix by the same ratio i.e. if the vehicle capacity is 70 units and the cost of 1 vehicle is 1000 euros, the price will be changed to 2000 if the vehicle capacity is changed to 140 units and will change to 500 euros if vehicle capacity becomes 35 units. This is done to keep the objective function result comparable, The results are shown in Figure 5.2.1.

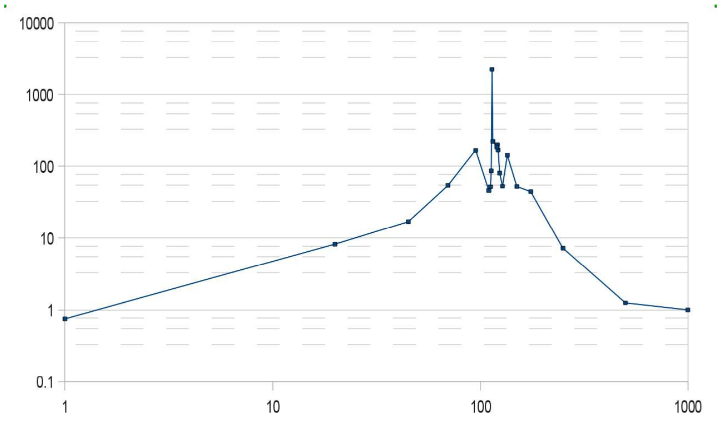

Figure 1: a Log-Log plot showing the effect of changing vehicle capacity φ on the time to find a solution using ILOG

5.3 Greedy algorithm

5.3.1 Basic outline

Greedy Algorithm (GA) is used for the evaluation of the same case presented for the Branch and Bound method in Section 5.2. Particular heuristic criteria are created to choose the new solution point for each step of the algorithm. Corresponding to each step of the greedy algorithm the objective function is calculated with the updates from the calculated solutions. To develop this method, a proper software has been developed using MATLAB, where the source code is reported in the appendix.

5.3.2 Selection criteria

Criteria are formulated for choosing pairs of sales points and factories in a way to fill the demand of sales point j from factory i. The greedy algorithm starts by identifying the sales point where there is a potential maximum lost sales cost, with the intent of improving the objective equation by reducing this cost. The candidate sales point j is selected with the criteria for a given planning time t is:

$$Max\left(\sigma_j^t L_j\right) \quad \forall j \tag{14}$$

After the selection of the sales point using formula [14], the next procedure is to select a factory to satisfy the corresponding demand at the given sales point. The factory selection is based on a combine minimum cost of setup cost $\varrho_i^t y_i^t$, holding cost $L_j \varsigma_j$, transportation cost $N_{ij}^t k_{ij}$ where each of this costs are homogenized per unit mathematically formulated in the Equation 15 . It should be noted that the y_i in this algorithm depends on two things, it will have the value of zero if the factory has not produced in this time period or the residual capacity it has (after being activated in the same period) is lesser than the demand. It also would have the value of 1 if is chosen to be activated, and/or the residual capacity in the factory is higher or equal to the demand wanted from a certain SP.

$$Min\left(\frac{\varrho_i(1-y_i)}{q_{ij}}+\frac{K_{ij}}{\varphi}+\frac{Cap_i-q_{ij}+\sigma_{t-1}}{Cap_i}\times h_i\right)\quad\forall i \tag{15}$$

With each step of calculation the residual demands at each factory are updated and if there is enough quantity of products in a given factory to fill a demand at a sales point those will be first used without another activation.
The following are the results found while updating the objective equation using the greedy approach. The results are presented for the given case study in each steps of the algorithm.

Table 4: Solution path followed by the greedy algorithm

Time period	step	Factory	Sales point	Amount shipped	Obj fun
1	1	3	5	100	42825
	2	3	4	80	41800
	3	3	3	60	38940
	4	2	2	35	42445
	5	2	1	35	43585
2	6	2	2	100	42825
	7	2	3	125	42610
	8	2	4	55	40700
	9	2	5	20	42500
	10	2	1	25	43600
3	11	4	5	110	43280
	12	1	1	150	44330
	13	1	2	120	40330
	14	2	3	100	40030
	15	3	4	60	43830

As it can be seen from Table 4 the final solution reached by the greedy algorithm is not the optimal one from the search space of the feasible solutions considered when compared with Branch & Bound presented in Section 5.2. With a simple visual inspection of each solution the optimality search in this space should have been terminated at the 3^{rd} step with a value of 38940. While further search by the greedy algorithm stops at a value worse than this value, which is 43830. The reason of such behavior of the algorithm is due to the hueristic approach used here, as it forces the greedy to satisfy the demand of each salespoint. Probably if the penalties of lost sales are higher the answer of the greedy would have been closer to the optimal solution found by B&B.

5.4 Greedy algorithm with Local Search

Finally, to improve the solution found by the Greedy Algorithm (GA) a Local Search (LS) algorithm was used. The approach used was the Hill climbing Local Search which only moves if a better position is found, i.e. if the objective function improves then we adopt the provided solution otherwise we go back to our previous step. The algorithm stops if there are no neighoubring moves with a better objective function. Also for LS, to develop this method, a proper software has been developed using MATLAB, where the source code is reported in the appendix.

In our problem, a solution (obtained by the GA) tells us how many products to send from Factory i to Salespoint j during the time period t. This quantity denoted Q_{ij}^t is an integer value that needs vehicles to be transported, and it required a number of vehicles to be shipped. Usually the Q obtained is fractions of Full Truck Load and thus we would have some unutilized space in the vehicles. The LS will round the Q sent to the next Full Truck Load (FTL), and only impose this value to the GA then trying to see what is the objective function value achieved from such modification. As described before we only will fix and use this value if the final output is better. The final solution reached by the Local Search algorithm is not the optimal one frome the search space of the feasible solutions considered, which is 40870 euros for the reference example.

5.5 Comparisons between methods

An extensive comparison between the methods has been done. The evaluated parameters considered in this comparison are the computing time, the number of iterations and the objective function value for the three different approached adopted for the problem, varying the number of time units. The results are reported in Table 5.

Table 5: Comparison results.

T	Computing time [s]			Iterations [#]				Objective function [Eur]		
	B&B	GA	LS	B&B	GA	LS [1]		B&B	GA	LS
3	0.250	0.062	0.338	12680	15	2	26	26935	43830	40870
4	0.500	0.070	0.382	20416	20	2	38	34405	54300	53590
5	2.000	0.069	0.489	65464	25	2	48	43895	69410	69190
6	2.250	0.084	0.607	64417	30	2	56	50382	87720	77570
7	10.750	0.092	0.798	312836	35	2	68	57549	102010	87890
8	54.410	0.088	0.937	1439136	40	2	78	67328	110410	100450
9	177.4	0.101	1.153	4402122	45	2	87	77097	120560	116270
10	222.0	0.096	1.377	6355959	50	2	98	83286	135160	125050
11	1497	0.110	1.812	34838282	55	2	108	92825	147580	138490
12	1537	0.114	1.866	40975461	60	2	116	102215	159700	153770

[1] The first column is the number of iterations made, while the second column is the total number of searches made.

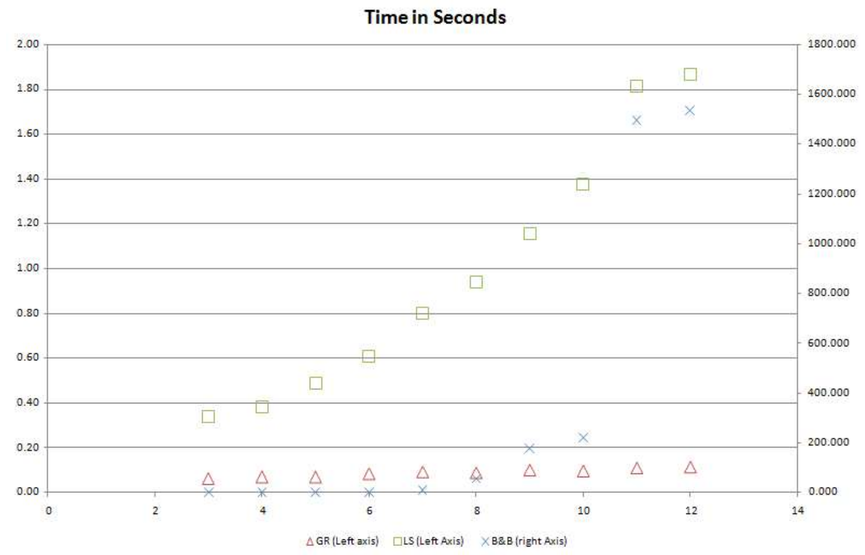

Figure 2: Computational time on time horizon.

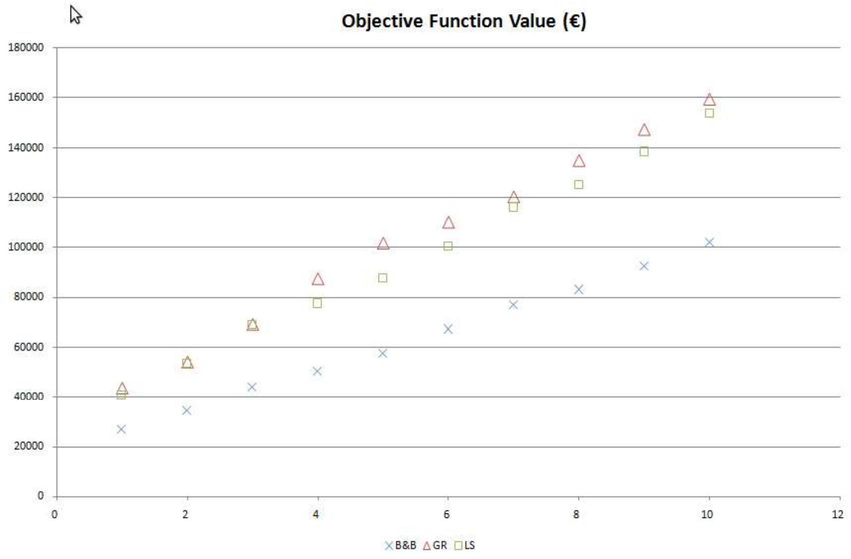

Figure 3: Objective function values on time horizon.

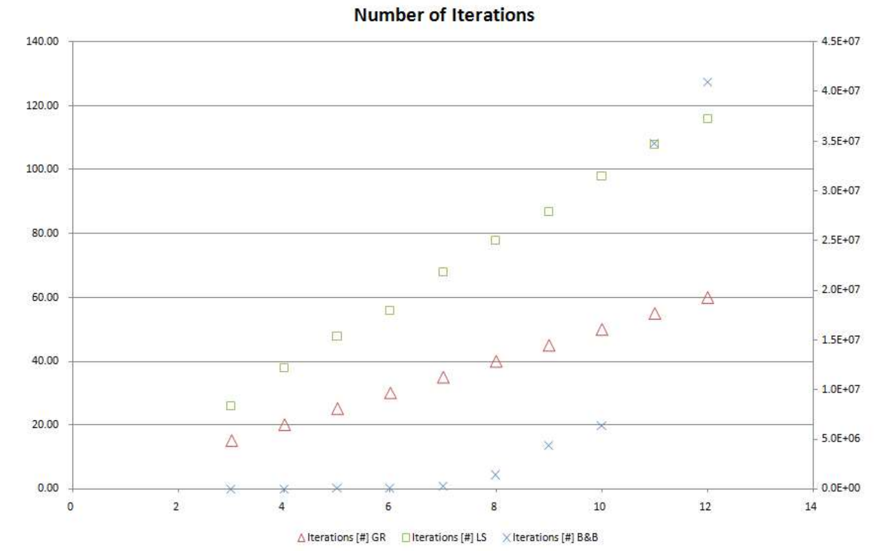

Figure 4: Number of iterations on time horizon.

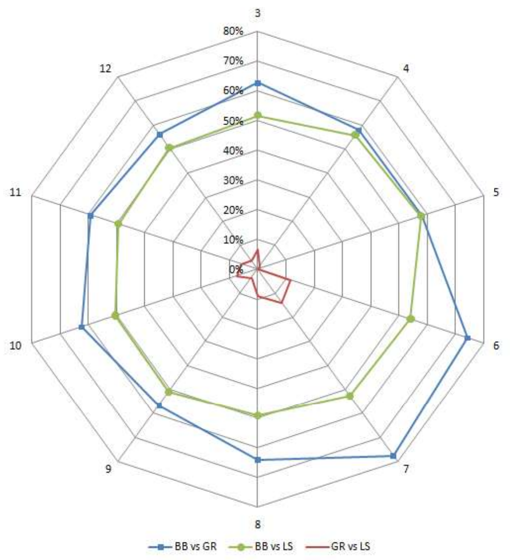

Figure 5: Comparison among the three different solutions found

In Figures 5.5, 5.5 and 5.5 are charted the graphs of the computation time, the objective function values and the number of iterations. It could be noticed from Figure 5.5 the exponential behavior of the solution time when using B&B, this due to the added complexity when the time horizon of the planning increase. This does not happen for the other two approaches LS and GA where they increase linearly with complextiy of the problem. Figure 5.5 shows that the solution found by B&B is always better than the others then comes the one found by LS and finally the GA which is an expected result. I should be noticed that LS effictiveness changes from one point to another, as seen in Figure 5.5

5.5.1 Using the Local Search Solution as an Initial Solution for B&B

From the previous section, we found that the difference between Local Search and B&B is high for a time horizon of 12 periods. In this subsection we will see the effect of using the solution found for Local Search as a starting solution for B&B. Applying this with ILOG, we found that the time reduction is insignificant, in fact after 18 minutes the algorithm did not find a solution and the program crashed. This is compared to a 25 min solution time by using null values for all variables. More experiments needs to be done to have a better insight though.

6 Conclusions and further development

In this work, we have formulated a two level SC distribution problem using MIP. Taking realistic assumptions into account, the problem considered is categorized as NP-hard problem. Three different approaches (B&B, Greedy Algorithm and Local Search) are used to find a solution for this problem applying an example case with different planning periods[t=3 to 12]. Results show that B&B solution is around 60% and 50% better than GA and LS respectively. Furthermore, the exponential behavior of the solution time was notices as planning period equals to 12, in which the other two methods solve 99.9% faster. Furthermore, the effect of the number of links was studied, it could be noticed that more than one factor affects solution time. Finally, The effect of starting from the GA solution was studied, although the program faced problems arriving at a point, but the time was not decreasing compared starting from a zero solution.

Further developments for the formulation include taking into account lead time between sales points and factories, introduction of transient points and different vehicle capacity. And for solution method, maybe comparing different heuristics for the Greedy Algorithm and the Local Search.

References

[1] A. Fugenschuh and A. Martin. Computational integer programming and cutting planes. *Discrete Optimization*, 12:69–122, 2001.

[2] J. Hoffmann. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. *Foundations of Intelligent Systems*, -:691–701, 2000.

[3] Y. Pochet and L.A. Wolsey. *Production planning by mixed integer programming.* Springer Verlag, 2006.

[4] L.A. Wosley. Integer programming, 1998.