Solving a Distribution Network Design Problem by means of Evolutionary Algorithms

Guillermo CABRERA[1], Stefanie NIKLANDER[2,3], Enrique CABRERA[4], Franklin JOHNSON[5]

[1] Pontificia Universidad Católica de Valparaíso, Chile
guillermo.cabrera@ucv.cl

[2] Universidad Autónoma de Chile, Chile

[3] Universidad Científica del Sur, Peru
sniklander@gmail.com

[4] Universidad de Valparaíso, CIMFAV, Chile
enrique.cabrera@uv.cl

[5] Universidad de Playa Ancha, Chile
franklin.johnson@upla.cl

Abstract: In this paper a simple and efficient evolutionary algorithm is implemented to solve a Distribution Network Design problem (DND). The DND problem that we address here integrates inventory policies with location/allocation decision making. This problem, also known as Inventory Location Modeling problem, is a complex combinatorial optimization problem that cannot be solved by exact methods as the number of decision variables increases. We compare our algorithm to previously implemented algorithms. Our evolutionary approach is shown to be very competitive in terms of both objective function value and execution time.

Keywords: Evolutionary Algorithm, Distribution Network Design, Logistics, Combinatorial Optimisation.

1. Introduction

The problem of locating/allocating customer to distribution centres is one of the most studied problems in logistics. Usually, after decision makers determine the locations to be installed, the inventory policy is defined. However, this sequential approach is sub-optimal in the sense that the inventory policy is restricted by the network design determined in the previous step. Thus, integrating location/allocation decisions and inventory policies lead to solutions that are more efficient as they consider the entire system as a whole.

On the one hand, the location/allocation problem consists of selecting specific sites at which to install plants, warehouses and/or distribution centres while assigning customers to service facilities and interconnecting facilities using flow assignment decisions [6, 10].

On the other hand, inventory policies determine tactical issues such as reorder point and order size, among others.

Figure 1 shows a schema of both the sequential and the integrated approaches. On the left hand side, the sequential approach is presented. As we can see, the first decision to be made is the location-allocation one. Once this decision is made, we can go to the next step which is to decide among the different inventory policies that are able to be implemented. Once this tactical decision is made, other operational issues are addressed. No change on previous decisions is allowed in this sequential approach. Thus, location-allocation decisions have a great impact on the final solution and, at the same time, both tactical and operational decisions depend on the distribution network that is defined during the first step of the approach. On the right hand side of the Figure 1, we have the integrated approach. Here we can note that information flow from one decision level to the other.

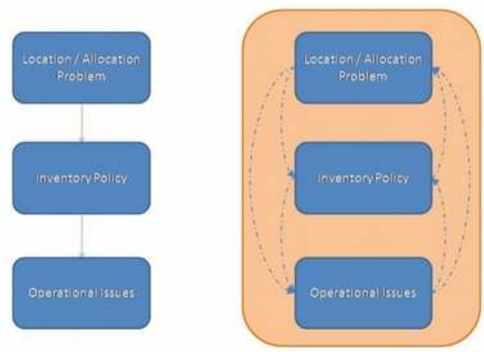

Figure 1. Traditional sequential approach (on the left) versus the integrated approach (on the right)

This means that strategic decisions are made taking into account the effects that such decisions provokes at tactical and operational levels. As a consequence, the final solution is more efficient in terms of its total system cost.

The remaining of this paper is as follows: In Section 2 the inventory location modelling problem that is addressed in this paper is presented. In Section 3 we introduce the evolutionary algorithm that is implemented here. In Section 4, the computational experiments performed in this paper are described and the obtained results are discussed. Finally, in Section 5, some conclusions and the future work are outlined.

2. Inventory-Location Modelling

Several inventory location models (ILM) have been proposed in the literature. Unlike facility location models, ILM does consider interactions between the facility location decision making process and the inventory policy to be implemented within each facility. Several authors have proposed different approaches to integrate both strategic and tactical level within the decision making process, see for instance [5, 11, 14, 17]. This integration of strategic and tactical decisions is very important as allows us to model the interactions that exist and the effect on the final distribution network design. One of these effects is the risk pooling effect. It states that the safety stock required by the whole distribution system will decrease as fewer warehouses are installed [11, 10, 15].

One key difference between the proposed approaches in the literature regards the inventory policy that is included within the decision making process. For instance, Daskin et. al. [8] and Shen et. al. [16] consider a continuous inventory policy, namely (Q, RP), and the well known uncapacitated facility location problem. They consider a safety stock at each facility, which does not interact with the other inventories from other facilities. A Lagrangian Relaxation algorithm is used to solve the ILM in [8] while a column generation approach is used in [16]. Using the same inventory policy, Miranda et al. [11] consider the order quantity for each warehouse as a decision variable and the capacitated facility location problem as a base framework. Finally, in Miranda et al 2008 [13] and Ozsen et al. [15] authors handle capacity constraints by using previous inventory-location models.

Unlike all the approaches mentioned above, Cabrera et al [7] have proposed a periodic review policy, known as (R, s, S). In their model, capacity constraint as well as the order size are handled as constraint and decision variable, respectively.

Multi-objective ILMs have been also proposed in the literature (see [1, 2]).

Since in general both continuous and integer decision variables are part of the models above, the ILM problem to be solved is usually described as a mixed integer one. Furthermore, the objective function that combines the location/allocation cost with the inventory costs is usually non-linear. Thus, solving this problem to optimality within a reasonable time is simply not possible as the number of decision variables becomes larger. Thus, several authors have considered heuristic methods to approximately solve this problem.

For instance, a Tabu search is used in [3] to solve a model that integrates production and distribution decisions by considering the capacity constraints of the plant. Evolutionary algorithms have been also used to solve the ILM problem (see Askin et. al. [4]). Recently, an approach that combines both exact algorithms and heuristic methods has been proposed [10]. This method belongs to the class of *matheuristic* methods.

In this paper, we solve the ILM problem introduced by Miranda and Garrido [12] by means of a simple but effective evolutionary algorithm. To the best of our knowledge no evolutionary algorithm has been used to solve such a problem in the literature before.

The mathematical model for the ILM that we aim to solve in this paper is as follows [12].

$$Min \sum_{i=1}^{N} F_i x_i + \sum_{i=1}^{N} \sum_{j=1}^{M} C_{ij} Y_{ij} + \sum_{i=1}^{N} CS_i \sqrt{V_i} + \sum_{i=1}^{N} CL_i \sqrt{D_i} \quad (1)$$

$$D_i = \sum_{j=1}^{M} d_j Y_{ij}, \quad \forall i = 1, \ldots, N \quad (2)$$

$$V_i = \sum_{j=1}^{M} v_j Y_{ij}, \quad \forall i = 1, \ldots, N \tag{3}$$

$$\sum_{j=1}^{M} d_j Y_{ij} = I_i^{cap} x_i \tag{4}$$

$$x_i, Y_{ij} = \{0,1\}, \quad \forall i = 1, \ldots, N, \quad \forall j = 1, \ldots, M \tag{5}$$

Equation (1) represents the total system cost. The first term is the cost of locating a specific plant/warehouse i, also called *setup cost*. The second term is the daily transport cost between warehouse and customers, where $C_{ij} = TH(TC_{ij} + RC_i)d_j$ and TH is the planning horizon, TC_{ij} is the transport allocation cost for allocating client j to warehouse i and RC_i is the transport cost associated to moving one unit of product from i to j. The parameter d_j is the mean demand of customer j. The third term corresponds to the cost of keeping a safety stock which minimises the stock out so we can guarantee a service level at least as good as $Z_{1-\alpha}$. Here we have that $CS_i = TH\,HC\,Z_{1-\alpha}\sqrt{LT_i}$ where HC_i corresponds to the holding cost of warehouse i and LT_i is the time between an order is placed and the products are available in our inventory. This time is also called *leadtime*. The variable V_i determines the total variance of the demand for warehouse i. Finally, the forth term in the objective function is the inventory cost, that is the cost of keeping products in stock and the administrative costs of putting an order to the suppliers, with $CL_i = TH\sqrt{2HC_iOC_i}$. Here OC_i is the order cost of warehouse i and D_i is the total demand of warehouse i. Constraint (4) ensures the capacity constraint of plant i, I_i^{cap}, will never be violated. As pointed out in [11], this is a very hard constraint that was relaxed in [12]. Finally, Equation (5) states integrality (0-1) for the binary variables Y_{ij} and x_i.

3. Proposed Approach

Heuristic methods are a common approach to solve hard combinatorial optimisation problems such as the ILM. In spite of the fact that heuristics do not guarantee optimality, the solutions provided by them can be considered good sub-optimal ones. In contrast exact methods guarantee optimality; however, they usually fail when dealing with medium- and large- sized problems. In this paper, an evolutionary algorithm is considered to solve the problem from Equations (1-5).

3.1 Evolutionary algorithm

As mentioned earlier, in this paper an evolutionary algorithm [9] is implemented to solver the ILM presented in the previous section. The evolutionary algorithm implemented here makes use of both crossover and mutation operators, and the fitness of each individual is set to be the same as the objective function value.

As in any other evolutionary algorithm, the individual representation needs to be defined. In this paper we represent an individual I_z as a vector of integer of the form

1	2	…	M_1	M

where I_i ranges from 1 to N, and corresponds to the warehouse assigned to customer i. Then a mutation operator will consist on changing one *gene* I_i to a new warehouse which could be either open or closed. Figure 2 shows an example of the mutation operator used in this paper.

5	4	4	5	1	1	3

↓

5	4	4	2	1	1	3

Figure 2. Example of the mutation operator used by the evolutionary algorithm.

As we can see in Figure 2, all but one gene keep their original value, while the customer $i=4$ is moved from warehouse 5 to warehouse 2, which was previously closed.

The crossover operator considered in this paper is as follows. Given two parents, we choose randomly a crossover point γ, which ranges from 2 to $M-1$. Then, the first offspring will have genes from 1 to γ equal to parent 1, and genes from $\gamma+1$ to M equal to parent 2. Offspring 2 will have genes from 1 to γ equal to parent 2 and genes from $\gamma+1$ to M equal to parent 2. Figure 3 shows this situation.

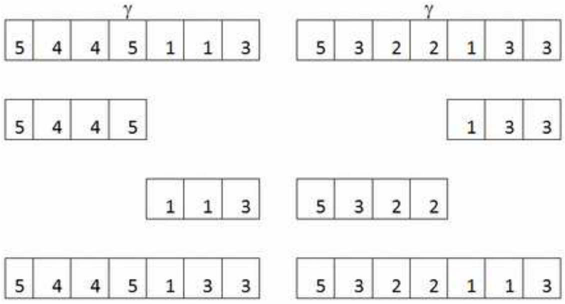

Figure 3. Example of the crossover operator used by the evolutionary algorithm.

Algorithm 1 shows the main steps of our approach.

Algorithm 1: Evolutionary Algorithm

Begin
$t=0$;
$P_t = initPopulation()$;
$evalPopulation(P_t)$;
$s_{best} = getBest(P_t)$;
while not termination criterion **do**
$P_{t+1} = nextGeneration(P_t)$;
$evalPopulation(P_{t+1})$;
if $s_{best} > getBest(P_{t+1})$
$s_{best} = getBest(P_{t+1})$;
End
$t = t+1$;
End
End
End

The evolutionary algorithm starts by generating the initial population (*initpopulation()*in algorithm 1). For each individual within the population the ILM is solved (*evalPopulation()* in algorithm 1). Then, the best individual within the population is saved as the best individual found so far by the evolutionary algorithm (*getBest()* in algorithm 1). After that, the next generation is created by using both algorithm (*getBest()* in algorithm 1). After that, the next generation is created by using bothmutation and crossover operators. A 90% of the individuals of the next generation are obtained by using crossover operator, while the remaining 10% is generated by using the mutation operator (*nextGeneration()* in algorithm 1). Parents in the crossover process are chosen by means of the tournament method, while the top 10% of the population is considered to be mutated. Once the next generation has been generated, we evaluate each individual. Sometimes individuals become infeasible. In such cases, a restoration phase is applied so the individual can be evaluated by the algorithm. If the best individual of the new population is better than the best individual found so far, then the best individual is updated and the new best individual is saved for next iterations. Termination criterion is the number of generation the algorithm must generate. At the end of the algorithm the individual with the best fitness found during the algorithm execution is set as the best solution found by the evolutionary algorithm.

4. Computational Experiments

In this subsection, we present the computational experiments carried out in this work. We use an Intel Core Duo processor CPU T2700, 2.33 GHz with 6 GB of RAM to run our experiments. Linux 14.02 was the operating system. The evolutionary algorithm was coded in JAVA 8 language using NetBeans IDE. The instance set used in this work was the same used in [12].

Four approaches are considered in the experiments. The first one is the sequential approach, denoted by *SDND*. This approach solves the corresponding capacity location problem and then implements the inventory policy in the obtained network configuration. The second and third approaches are the ones implemented in [12]. These approaches are denoted by *Lingo* and *LR,* respectively. Finally, the last approach is our evolutionary algorithm, denoted by *EA*.

The base instance that is presented in the results section is denoted by X . Just as in [12], we create four additional instances that are based on instance X . These additional instances are denoted by $X_{-50\%HC}$, $X_{-25\%HC}$, $X_{+25\%HC}$ and $X_{+50\%HC}$. These instances modify the holding cost of the original $X_{\square}$ instance. For instance, $X_{-50\%HC}$ means holding cost HC has been reduced in 50%.

4.1 Results

In this section we present a summary of the obtained results. Table 1, 2, 3 and 4 show the obtained results when the service level, $Z_{1-\alpha}$ is set to 50%, 75%, 90% and 97.5%, respectively.

First column denotes the instance name. Then, for each approach the cost and the comparison with the best solution found for the corresponding instance are shown.

As we can see in Table 1, as the problem becomes more complex (the number of customers increases) our approach obtains better results. Moreover, as highlighted by [12], as the holding costs *HC* become more important, the integrated approach obtains better results than the sequential approach.

When the service level is set to 0.5, that is the probability of stock-out is 50%, the importance of the integration is not much. As Table 1 shows, the best average value is obtained by the exact algorithm *lingo* and, in the second position we can find our evolutionary algorithm.

Although a good result if compared to the *Lagrangean relaxation* or *SDND* approaches, the evolutionary algorithm does not find the best solution in any of the instances when $Z_{1-\alpha}=0.5$.

In Table 2, the results obtained when the service level is set to $Z_{1-\alpha}=0.75$ are presented. As we can see, deficiencies of the sequential approach become more evident as the service level is higher. Moreover, in this case, the evolutionary algorithm finds, in average, better solutions than all the other algorithms, finding a better solution than all the other algorithms for two of the instances. It is interesting to note than the exact algorithm (*lingo*) consistently finds better solutions for those instances where the holding cost are more important.

Table 1. Obtained results for $Z_{1-\alpha}=0.5$

	SDND		**Lingo**		**LR**		**EA**	
Instance	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)
$X_{-50\,HC}$	600891	0,00%	600891	0,00%	605891	0,83%	605716	0,80%
$X_{-25\,HC}$	624418	0,00%	624418	0,00%	624521	0,02%	625370	0,15%
X	644260	0,76%	639400	0,00%	644260	0,76%	641784	0,37%
$X_{+25\,HC}$	661725	1,19%	653912	0,00%	661725	1,19%	656315	0,37%
$X_{+50\,HC}$	677493	1,57%	667009	0,00%	677297	1,54%	669454	0,37%
	Average	**0,71%**		**0,00%**		**0,87%**		**0,41%**

Table 2. Obtained results for $Z_{1-\alpha}=0.75$

	SDND		**Lingo**		**LR**		**EA**	
Instance	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)
$X_{-50\,HC}$	719798	5,07%	702188	2,50%	719552	5,04%	685035	0,00%
$X_{-25\,HC}$	757348	5,14%	733394	1,81%	754136	4,69%	720337	0,00%
X	795126	3,96%	764802	0,00%	776477	1,53%	768528	0,49%
$X_{+25\,HC}$	832805	3,62%	803743	0,00%	809202	0,68%	811317	0,94%
$X_{+50\,HC}$	870593	4,46%	833453	0,00%	843010	1,15%	852687	2,31%
	Average	**4,45%**		**0,86%**		**2,62%**		**0,75%**

In Table 3, the results obtained when the service level is set to $Z_{1-\alpha}=0.90$ are presented. Again, the sequential approach obtains the worst values while, in average, the *lingo* algorithm is the best approach. In second place we find, again, our evolutionary algorithm. In this case, our algorithm found better results than *lingo* for those instances where the holding costs are reduced. Furthermore, *Lagrangean relaxation* outperforms our evolutionary algorithm for the last two instances, where the holding costs are amplified.

Finally, we present in Table 4, the results obtained when the service level is set to $Z_{1-\alpha}=0.975$ are presented.

As in previous experiments, the sequential approach is outperformed by all the other approaches. Our evolutionary algorithm finds better solutions than all the other approaches for the first two instances, while *lingo* algorithm finds better solutions for the last three instances. Unlike in the case where $Z_{1-\alpha}=0.90$ (see Table 3), in this case our evolutionary algorithm found consistently better solutions than the *Lagrangean relaxation* approach for all the instances.

Unlike exact methods, the performance of the evolutionary algorithm implemented in this paper is not impaired as the problem size get larger.

Table 3. Obtained results for $Z_{1-\alpha}=0.90$

	SDND		**Lingo**		**LR**		**EA**	
Instance	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)
$X_{-50\ HC}$	787831	5,99%	767490	3,25%	783668	5,43%	743304	0,00%
$X_{-25\ HC}$	859201	7,08%	818078	1,95%	832812	3,79%	802420	0,00%
$X_{\square}$	931003	5,68%	880998	0,00%	898893	2,03%	882627	0,18%
$X_{+25\ HC}$	1002618	6,97%	937329	0,00%	961110	2,54%	971854	3,68%
$X_{+50\ HC}$	1074440	8,11%	993807	0,00%	1025370	3,18%	1055849	6,24%
	Average	**6,76%**		**1,04%**		**3,39%**		**2,02%**

Table 4. Obtained results for $Z_{1-\alpha}=0.975$

	SDND		**Lingo**		**LR**		**EA**	
Instance	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)	Cost	Save (%)
$X_{-50\ HC}$	863872	6,72%	828161	2,31%	860043	6,25%	809453	0,00%
$X_{-25\ HC}$	973042	6,82%	914062	0,35%	935273	2,68%	910895	0,00%
$X_{\square}$	1082874	8,24%	1000429	0,00%	1032900	3,25%	1028371	2,79%
$X_{+25\ HC}$	1192421	9,74%	1086592	0,00%	1130880	4,08%	1119760	3,05%
$X_{+50\ HC}$	1302283	11,02%	1172978	0,00%	1218510	3,88%	1213294	3,44%
	Average	**8,51%**		**0,53%**		**4,03%**		**1,86%**

As future work, other evolutionary algorithms, such as e.g. cultural algorithms, can be implemented to solve the ILM problem addressed in this paper.

Moreover, the problem can be converted into a multi-objective one and other ILM might be also considered to be solved with the evolutionary algorithm that is implemented here.

4. Conclusions and Future Work

In this paper we solve an ILM problem by means of a simple evolutionary algorithm. The evolutionary algorithm is shown to be competitive when compared to both exact methods and other heuristic methods. Moreover, distribution network designs obtained by solving the integrated ILM are consistently better than those obtained when solving the problem using the sequential approach in terms of the objective function value.

REFERENCES

1. AHMADI, G., S. A., TORABI, R. TAVAKKOLI-MOGHADDAM, **A Bi-Objective Location-Inventory Model with Capacitated Transportation and Lateral Trans-shipments**, International Journal of Production Research, 2015, pp. 1-22.
2. ARABZAD, S. M., M. GHORBANI, R., TAVAKKOLI-MOGHADDAM, **An Evolutionary Algorithm for a New Multi-Objective Location-Inventory Model in a Distribution Network with Transportation Modes and Third-Party Logistics Providers**, International Journal of Production Research, vol. 53(4), 2015, pp. 1038-1050.
3. ARMENTANO, V. A., A. L. SHIGUEMOTO, L. A., LOKKETANGEN, **Tabu Search with Path Relinking for an Integrated Production-Distribution Problem**, Computers and Operations Research, vol. 38(8), 2011, pp. 1199-1209.
4. ASKIN, R. G., I., BAFFO, M., XIA, **Multicommodity Warehouse Location and Distribution Planning with Inventory Consideration**. International Journal of Production Research, vol. 52(7), 2014, pp. 1897-1910.
5. BRADLEY, J. R., B. C. ARNTZEN, **The Simultaneous Planning of Production, Capacity, and Inventory in Seasonal Demand Environments**, Operations Research, vol. 47(6), 1999, pp. 795-806.
6. CABRERA, G. G., E. CABRERA, R. SOTO, L. J. M. RUBIO, B. CRAWFORD, F. PAREDES, **A Hybrid Approach Using an Artificial Bee Algorithm with Mixed Integer Programming Applied to a Large-Scale Capacitated Facility Location Problem**, Mathematical Problems in Engineering, vol. 2012, Article ID 954249, 14 pages, 2012.
7. CABRERA, G., G., P. A. MIRANDA, R. E., CABRERA, R. SOTO, L. J. M. RUBIO, B. CRAWFORD, F. PAREDES, **Solving a Novel Inventory Location Model with Stochastic Constraints and Inventory Control Policy, Mathematical Problems in Engineering**, vol. 2013, Article ID 670528, 12 pages, 2013.
8. DASKIN, M. S., C. R. COULLARD, Z. M. SHEN, **An Inventory Location Model: Formulation, Solution Algorithm and Computational Results**, Annals of Operations Research, vol. 110, 2002, pp. 83-106.
9. HOLLAND, J. H., **Adaptation in Natural and Artificial Systems**, University of Michigan Press, Ann Arbor, Michigan, USA, 1975.
10. LAGOS, C., F. PAREDES, S. NIKLANDER, F. E. CABRERA, **Solving a Distribution Network Design Problem by Combining Ant Colony Systems and Lagrangian Relaxation**, Studies in Informatics and Control, vol. 24(3), 2015, pp. 251-260.
11. MIRANDA, P. A., R. A. GARRIDO, **Incorporating Inventory Control Decisions into a Strategic Distribution Network Design Model with Stochastic Demand. Transportation Research Part E**, Logistics and Transportation Review, vol. 40(3), 2004, pp. 183-207.
12. MIRANDA, P. A., R. A. GARRIDO, **A Simultaneous Inventory Control and Facility Location Model with Stochastic Capacity Constraints**, Networks and Spatial Economics, vol. 6, 2006, pp. 39-53.

13. MIRANDA, P. A., R. A. GARRIDO, **Valid Inequalities for Lagrangian Relaxation in an Inventory Location Problem with Stochastic Capacity. Transportation Research Part E**: Logistics and Transportation Review, vol. 44(1), 2008, pp. 47-65.

14. MOURITS, M., J. M. EVERS, **Distribution Network Design: An Integrated Planning Support Framework**, International Journal of Physical Distribution & Logistics Management, vol. 25, 1995, pp. 43-57.

15. OSZEN, L., C. R. COULLARD, M. S. DASKIN, **Capacitated Warehouse Location Model with Risk Pooling**, Naval Research Logistics, vol. 55(4), 2008, pp. 295-312.

16. SHEN, Z. M., C. R. COULLARD, M. S. DASKIN, **A Joint Location Inventory Model**, Transportation Science, vol. 37(1), 2003, pp. 40-55.

17. SIMCHI-LEVI, D., Y. ZHAO, **The Value of Information Sharing in a Two Stage Supply Chain with Production Capacity Constraints**, Naval Research Logistics, vol. 50(8), 2003, pp. 888-916.