ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-563

Genetic Optimization of the Multi-Location Transshipment Problem with Limited Storage Capacity

Nabil Belgasmi and **Lamjed Ben Saïd** and **Khaled Ghédira**[1]

Abstract. Lateral Transshipments afford a valuable mechanism for compensating unmet demands only with on-hand inventory. In this paper we investigate the case where locations have a limited storage capacity. The problem is to determine how much to replenish each period to minimize the expected global cost while satisfying storage capacity constraints. We propose a Real-Coded Genetic Algorithm (RCGA) with a new crossover operator to approximate the optimal solution. We analyze the impact of different structures of storage capacities on the system behaviour. We find that Transshipments are able to correct the discrepancies between the constrained and the unconstrained locations while ensuring low costs and system-wide inventories. Our genetic algorithm proves its ability to solve instances of the problem with high accuracy.

1 INTRODUCTION

Practical optimization problems especially supply chain optimization problems, usually have a complex structure. That is the same in a lot of transport or production related fields [1]. Physical pooling of inventories has been widely used in practice to reduce cost and improve customer service [2]. Transshipments are recognized as the monitored movement of material among locations at the same echelon. It affords a valuable mechanism for correcting the discrepancies between the locations' observed demand and their on-hand inventory. Subsequently, Transshipments may reduce costs and improve service without increasing the system-wide inventories.

The study of multi-location models with Transshipments is an important contribution for mathematical inventory theory as well as for inventory practice. The idea of lateral Transshipments is not new. The first study dates back to the sixties. The two-location-one-period case with linear cost functions was considered by [3]. [4] studied with N-location-one-period model where the cost parameters are the same for all locations. [5] incorporated non-negligible replenishment lead times and Transshipment lead times among stocking locations to the multi-location model. The effect of lateral Transshipment on the service levels in a two-location-one-period model was studied by [6]. The common problem tackled by these models is the determination of the optimal replenishment decision which minimizes the aggregate cost of the system. Most of the studies lead to optimal solutions since they investigate simple models easily solved by mathematical techniques (see [4]], [7]). However, an optimal replenishment decision for a general multi-location inventory system cannot be solved in analytical way. Few operational research methods were applied to find out near-optimal solutions. The gradient-based IPA method was successfully used for both capacitated Transshipment and production problems [8]. The use of IPA to solve real-world problems is not always possible since many conditions should be satisfied to ensure the unbiasedness of its estimator [9]. Evolutionary optimization may provide a powerful methodology for solving such complex problems without need of prior knowledge about their analytical properties. The contribution of this paper is twofold. We, first, incorporate storage capacity constraints to the traditional Transshipment model which leads to a better modelling of real-world situations. Second, we investigate the applicability of real-coded evolutionary algorithms to the optimization of inventory levels and costs. This provides insights to tackle other extensions of the basic Transshipment problem with evolutionary optimization methods.

The remainder of this paper is organized as follows. In Section 2, we formulate the proposed Transshipment model. In Section 3, we present the main concepts of the evolutionary optimization; we describe the new crossover operator and our evolutionary modelling of the problem. In Section 4, we show our experimental results. In Section 5, we state our concluding remarks.

2 THE PROBLEM

2.1 Model description

We consider the following real life problem where we have n stores selling a single product. The stores may differ in their cost and demand parameters. The system inventory is reviewed periodically. At the beginning of the period and long before the demands realization, replenishments take place in store i to increase the stock level up to S_i. The storage capacity of each location is limited to $S_{max,i}$. In other way, the replenishment quantities should not exceed $S_{max,i}$ inventory units. This may be due to expensive fixed holding costs, or to the limited physical space of the stores. Thus, the inventory level of store i will be always less or equal to $min(S_i, S_{max,i})$. After the replenishment, the observed demands D_i which represents the only uncertain event in the period are totally or partially satisfied depending on the on-hand inventory of local stores. However, some stores may be run out of stock while others still have unsold goods. In such situation, it will be possible to move these goods from stores with surplus inventory to stores with still unmet demands. This is called lateral Transshipment within the same echelon level. It means that stores in some sense share the stocks. The set of stores holding inventory I^+ can be considered as temporary suppliers since they may provide other stores at the same echelon level with stock units. Let t_{ij} be the Transshipment cost of each unit sent by store i to satisfy a one-unit unmet demand at store j. In this paper, the Transshipment lead time is considered negligible. After the end of the Transshipment process, if store i still has a surplus inventory, it will be penalized by a per-unit holding cost of h_i. If store j still has unmet demands, it will be penalized by a per-unit

[1] Ecole Nationale des Sciences de l'Informatique,
Email : khaled.ghedira@isg.rnu.tn

shortage cost of p_j. Fixed cost Transshipment costs are assumed to be negligible in our model. [2] proved that, in the absence of fixed costs, if Transshipments are made to compensate for an actual shortage and not to build up inventory at another store, there exists an optimal base stock policy S^* for all possible stationary policies. To see the effect of the fixed costs on a two-location model formulation, see [10]. The following notation is used in our model formulation:

n	Number of stores
S_i	Order quantities for store i
S	Vector of order quantities, $S = (S_1, S_2, \ldots, S_n)$ (Decision variable)
$S_{max,}$	Maximum storage capacity of store i
S_{max}	Vector of storage capacities, $S_{max} = (S_{max\ 1}, S_{max,2}, \ldots, S_{max\ n})$
D_i	Demand realized at i
D	Vector of demands, $D = (D_1, D_2, \ldots, D_n)$
h_i	Unit inventory holding cost at i
p_j	Unit penalty cost for shortage at j
t_{ij}	Unit cost of Transshipment from i to j
T_{ij}	Amount transshipped from i to j
I^+	Set of stores with surplus inventory (before Transshipment)
I^-	Set of stores with unmet demands (before Transshipment)

2.2 Modelling assumptions

Several assumptions are made in this study to avoid pathological cases:

- **Assumption 1 (Transshipment policy):** The Transshipment policy is stationary, that is, the Transshipment quantities are independent of the period in which they are made; they depend only on the available inventory after demand observation. In this study, we will employ a Transshipment policy known as complete pooling. This Transshipment policy is described as follow [11]: "the amount transshipped from one location to another will be the minimum between (a) the surplus inventory of sending location and (b) the shortage inventory at receiving location". The optimality of the complete pooling policy is ensured under some reasonable assumptions [6].
- **Assumption 2 (Lead time):** Transshipment lead times are negligible. At the end of every period, optimal Transshipment quantities are computed. We assume that they are immediately shipped to their destination without making customers wait for long time.
- **Assumption 3 (Replenishment policy):** At the beginning of every period, replenishments take place to increase inventory position of store i up to $min(S_i, S_{max,i})$ taking into account the remaining inventory of the previous period. The optimality of the order-up-to policy in the absence of fixed costs is proven in [2].

2.3 Model formulation

Cost function: Since inventory choices in each store are centrally coordinated, it would be a common interest among the stores to minimize aggregate cost. At the end of the period, the system cost is given by:

$$C(S,D)=\sum_{i\in I^+} h_i(S_i-D_i)+\sum_{j\in I^-} p_j(D_j-S_j)-K(S,D) \quad (1)$$

The first and the second term on the right hand side of (1) can be respectively recognized as the total holding cost and shortage cost before the Transshipment. However, the third term is recognized as the aggregate Transshipment profit since every unit shipped from i to j decreases the holding cost at i by h_i and the shortage cost at j by p_j. However, the total cost is increased by t_{ij} because of the Transshipment cost. Due to the complete pooling policy, the optimal Transshipment quantities T_{ij} can be determined by solving the following linear programming problem:

$$K(S,D)=\max_{T_{ij}} \sum_{i\in I^+}\sum_{j\in I^-}(h_i+p_j-\tau_{ij})T_{ij} \quad (2)$$

Subject to

$$\sum_{j\in I^-} T_{ij} \le S_i - D_i \quad ,\forall i\in I^+ \quad (3)$$

$$\sum_{i\in I^+} T_{ij} \le D_j - S_j \quad ,\forall j\in I^- \quad (4)$$

$$T_{ij} \ge 0 \quad (5)$$

In (2), problem K can be recognized as the maximum aggregate income due to the Transshipment. T_{ij} denotes the optimal quantity that should be shipped from i to fill unmet demands at j. Constraints (3) and (4) say that the shipped quantities cannot exceed the available quantities at store i and the unmet demand at store j. Since demand is stochastic, the aggregate cost function is built as a stochastic programming model which is formulated in (6). The objective is to minimize the expected aggregate cost per period.

$$\min_S \mathrm{E}(C(S,D))=\min_S \mathrm{E}\left(\sum_{i\in I^+} h_i(S_i-D_i)+\sum_{j\in I^-} p_j(D_j-S_j)-K(S,D)\right) \quad (6)$$

Subject to

$$S_i \le S_{\max,i}, \forall i=1\ldots n \quad (7)$$

where the first two terms denotes the expected cost before the Transshipment, called Newsvendor[2] cost, and the third term

[2] The newsvendor model is the basis of most existing Transshipment literature. It addresses the case where Transshipments are not allowed.

denotes the expected aggregate income due to the Transshipment. This proves the important relationship between the newsvendor and the Transshipment problem. By setting very high Transshipment costs, i.e. $t_{ij} > h_i + p_j$, no Transshipments will occur. Problem K will then return zero. Thus, our model can deal with both Transshipment and newsvendor cases.

Cost function properties: The cost function is stochastic because of the demand randomness modelled by the continuous random variables D_i with known joint distributions. Thus we must compute the expected value of the cost function. An analytical tractable expression for problem K given in (2) exists only in the case of a generalized two-location problem or N-location with identical cost structures [4]. In both cases, the open linear programming problem K has an analytical solution. But in the general case (many locations with different cost structures), we can use any linear programming method to solve problem K. In this study, we used the Simplex Method. The mentioned properties of our problem are sufficient to conclude that it is not possible to compute the exact expected values of the stochastic function given in (6). The most common method to deal with noise or randomness is re-sampling or re-evaluation of objective values [12]. With the re-sampling method, if we evaluate a solution S for N times, the estimated objective value is obtained as in equation (8) and the noise is reduced by a factor of $\sqrt{N}$. For this purpose, draw N random scenarios $D^1,\ldots,D^N$ independently from each other (in our problem, a scenario D^k is equivalent to a vector demand $D^k=(D^1{}_1,\ldots,D^N{}_N)$. A sample estimate of $f(S)$, noted $E(f(S,D))$, is given by

$$\mathrm{E}\big(f(S,D)\big) \approx \overline{f}(S) = \frac{1}{N}\sum_{k=1}^{N} f(S,D^k) \Rightarrow \overline{\sigma} = \sqrt{Var[\overline{f}(S)]} \approx \frac{\sigma}{\sqrt{N}} \tag{8}$$

3 EVOLUTIONARY OPTIMIZATION

3.1 Main concepts

We refer to evolutionary algorithms as methods that handle a population of solutions, iteratively evolve the population by applying phases of *self-adaptation* and *co-operation* and employ a *coded representation* of the solutions. The most suitable evolutionary algorithm to solve optimization problems in continuous domains are Evolutionary Strategies (ES) [13], Genetic Algorithms (GA) [14] with real coding and evolutionary programming [15]. GAs are search methodology invented by Holland [15], which is inspired by the natural genetic theory. They are regarded as methods that are suited for exploring large solution spaces. It is a very effective method for solving real-world problems this success is its simplicity and performance. The main idea of this technique is to generate diverse chromosomes and select the most appropriates to continue. We have an initial population of chromosomes which are produced randomly or by a particular scheme. Then, iteratively, we generate new generations of population out of the previous ones using mutation, crossover and selection. Mutation is designed to generate a new chromosome out of an existing one by randomly changing it. In the crossover two existing chromosomes are combined to generate new chromosomes. Selection will ensure the formation of the new population from the previous population. By applying the mentioned operations, the average fitness of the population will tend to increase over the algorithm lifetime. In many practical problems, chromosomes are coded as real numbers. We call the GA working with real parameters in its chromosome RCGA (Real Coded Genetic Algorithm). The general structure of a GA is:

```
Genetic algorithm
Begin
  t:=0
  Initialize P(t)
  Evaluate P(t)
  while (not Stop-criterion) do
    t := t + 1
    Select POP(t) from P(t-1)
    Crossover P(t)
    Mutate P(t)
    Evaluate P(t)
  End-While
End.
```

Where t is the current generation, and $P(t)$ is the current population.

3.2 Solution methodology

In our study, a real-coded GA is used to search for optimal replenishment decisions S^*, with respect to the storage capacity constraints. In this section, we describe our evolutionary modelling of the constrained multi-location Transshipment problem.

Structure of the Individual and population size: Each individual consists of a vector of n genes. It encodes a replenishment decision S. A gene is a positive real parameter representing an order quantity S_i. It is easy to see that a population represents a set of replenishment decisions that moves toward regions of the search space that have better fitness values (lower costs). The population size is less than 30 individuals.

Fitness evaluation: With respect to the re-sampling method given in (8), we should evaluate each individual N times in order to compute its fitness value. However, this may lead to individuals with different variances, which makes the selection of good individuals not accurate. Thus, in order to get a population with a common estimation Error Rate ER, we repeat the evaluation of each individual until its error estimation rate would be less than ER. We define the error estimation rate as the fraction of the estimated standard deviation and the expected mean of the sampled function at the given design S,

$$ER(S) = \frac{\overline{\sigma}(S)}{\overline{f}(S)} \tag{9}$$

Recall that $ER(S)$ is null when the approximated standard deviation is null. This is the case when the sample size is too large (9). Using the ER measure facilitates the supervision of the accuracy of explored regions of the search space, since neither the standard deviation nor the expected cost is known in advance. We will use ER varying between 0.01 and 2.

Initialization: In most of the search algorithms, the initialization method is very important. We have opted for two initialization procedures. The first consists of generating uniformly distributed values for each gene within the domain $[0, min(S_i, S_{max,i})]$. The second consists of analytical solving of the newsvendor version of our problem. Then, we initialize each gene with a random value close to the optimal computed solution with respect to the storage capacities.

Selection: After evaluating the fitness of each individual, we must select the fittest ones to reproduce and form the population of the next generation. In our case, the best individuals represent the set of replenishment decisions *{S*}* that ensure low aggregate costs. Many selection methods were studied and used for solving problems. We have chosen a deterministic selection procedure which consists of sorting the individuals and copying the best *10%* of them to the mating pool. This protects the best individuals and let them survive until the birth of stronger offspring.

Crossover: Mating is performed using crossover to combine genes from different parents to produce new children. We have chosen the binary tournament selection to pick out parents for reproduction. Tournament selection runs a tournament between two randomly chosen individuals and selects the winner (individual with best fitness value). Many crossover techniques were studied in evolutionary optimization. We tested 3 existing crossover operators. Let *A* and *B* be two selected parents, and a a real number uniformly generated between 0 and 1;

- **Single-point crossover:** the chromosomes of the parents are cut at a randomly chosen point and the resulting fragments are swapped.
- **Uniform crossover:** each gene of the offspring X is selected randomly from the corresponding genes of the parents.
- **Convex crossovers:** offspring X = a.A + (1-a).B

Moreover, we proposed a new crossover operator called *Gradient-descent crossover (GRD-Crossover)* since it creates an offspring following a quasi-descent direction. The first new offspring *X* is obtained by applying a convex crossover (*X* is inside the segment *[AB]*). The second offspring *Y* depends on the fitness values of the parents. Let C_A and C_B be fitness values of *A* and *B* and assume that $C_B = C_A$. We can suppose that if *Y* will be in the same direction of the path linking solution *A* to *B,* then it may be better than its parents. More properly, *X* and *Y* are created as below:

- X = a.A + (1 – a).B
- Y = B – *? .(B – A)*

Where a is a real number uniformly generated between 0 and 1; *?* is a positive uniform random variable that has the same sign as $(C_B – C_A)$. We implemented all these crossovers and showed that the *GRD-Crossover* performs well in term of convergence and accuracy.

Mutation: Mutation is realized by adding to each gene S_i a normally distributed random number centred on 0. This operator alters genes of the selected individuals with a given mutation probability. Because we are dealing with real-valued definition domains *(e.g.* $[0, min(S_i, S_{max,i})]$*)*, all offspring genes that are out of its domains are scaled down as follow: $S_i := min(S_i, S_{max,i})$.

4 OPTIMIZATION RESULTS

In this section, we report on our numerical study. We first analyze the shape of the constrained cost function for a given system setting. We illustrate the spread of the individuals in the first and the tenth generations of the GA. We compare our GRD-Crossover with other crossovers and show its ability to perform well and to provide near-optimal solutions. Finally, we analyze the impact of the incorporation of storage capacity in the basic Transshipment model.

4.1 Case study

Our first exemplary inventory model consists of 2 locations with the following parameters: $h_i=\$1$, $p_i=\$4$, $t_{ij}=\$0.5$ and $D_i=N(100,20)$. Location (2) has no storage capacity constraints ($S_{max,2}=8$). However, location (1) storage capacity is limited to $S_{max,1} = 80$. We generated 30.000 samples of the cost function with a fixed error rate *ER=1%.* The average number of evaluations is 450.000. Obviously, an individual consists of 2 genes only, each for one location. The evolutionary optimization process was started with the following parameters:

- Population size = 30
- Number of generations = 40
- Crossover rate = 85%
- Mutation rate = 15%
- Error rate = 1%

4.2 Experimental design

To show the flexibility of our model, we have studied a 4-location Transshipment system with 7 storage capacities. In all designs, holding costs are equal to \$1, shortage costs are equal to \$4, Transshipment costs are equal to \$0.5 and demands are normally distributed: N (100, 20). Table 1 summarizes the designs characteristics.

Table1. RCGA parameters

Sys	C-0	C-1	C-2	C-3	C-4	C-5	C-6	C-7
$S_{max,1}$	8	8	100	80	60	40	20	0

In system C-0, no material movement is allowed among locations. It represents 4 independent newsvendor problems. System C-1 refers to the basic Transshipment problem with no storage limits. In systems C2-7, only location (1) faces different storage constraints. All the other locations have no such storage constraints.

System-wide inventories considerably decrease in comparison to independent newsvendors system. Figure 1 reveals also an important property of multi-location systems with storage capacity constraints, that is the ability of the locations to face heavy storage constraints ($S_{max,1}$ = 0). Solidarity and cooperation of some system locations significantly fix the aggregate cost. When analyzing the optimal costs of all settings, we remark that whatever the hardness of the storage capacity (varying from 8 to 0), costs and system-wide inventories in systems where Transshipments are allowed (C-1-7) are less then newsvendor.

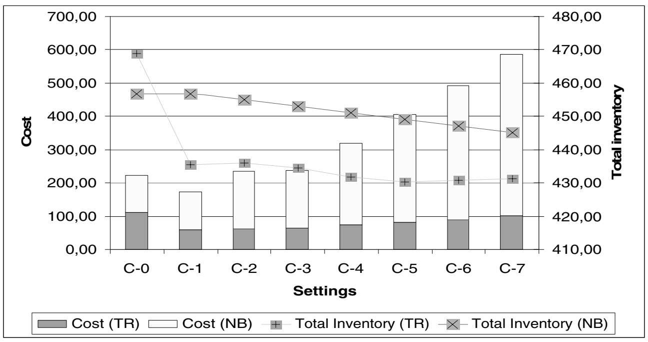

Figure 1. Cost under different systems

4.3 Validation with a benchmark

We validate our RCGA using an illustrative example from [4] where optimal solutions are available. Recall that the system consists of 4 locations having identical cost structures with a holding cost of $1 per unit, a shortage cost of $4 per unit, and Transshipment cost of $0.10 per unit. There are no storage capacity constraints. Thus, our purpose is to compare the solution given by our RCGA using different crossovers to the optimal solution computed analytically. This can be done by setting infinite storage capacity limits ($S_{max,i} = 8$).

In figure 2, we found that GRD-Crossover is better than all the other experimented crossovers. It has an important role in fine-tuning the individuals at the last generations. It performs better than the Convex-crossover though it is partially based on a convex exploitation of the selected parents. Figure 3 shows that best solutions given by the RCGA has a big variance ($94<S1<130$, $209<S2<274$, $155<S3<198$ and $148<S4<218$) whereas the resulting costs are approximately equal ($C = \{113.51, 113.90, 113.80, 115.42\}$). Recall the optimal solution is $S^*=(109, 222.5, 163.5, 192.5)$ with a minimal cost of $C^* = 113.49$. This leads to the conclusion that the approximation of the optimal cost value with our RCGA is satisfactory even though the approximation of the optimal order quantities has a great variance.

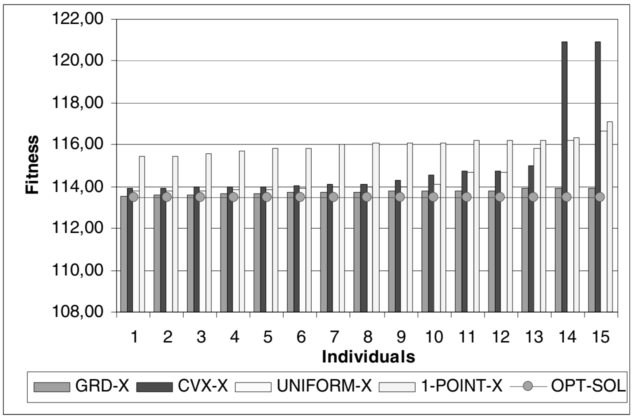

Figure 2. Best fitness of the last generation individuals under multiple crossovers

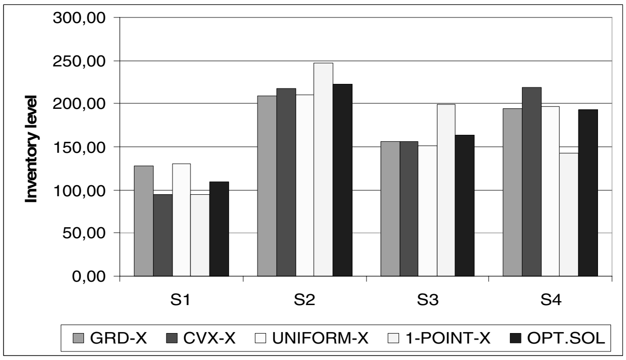

Figure 3. Optimal and near-optimal solutions under multiple crossovers

5 CONCLUSION

In this paper, we considered a multi-location Transshipment model with limited storage capacity. The objective is to minimize the aggregate cost function where decision variables are the constrained order-up-to quantities. We modelled the optimal redistribution of inventory in an arbitrary period as a linear programming problem based on the complete pooling policy. We employed a real-coded GA to solve the problem. A new crossover operator based on a simple approximation of the gradient descent is proposed and tested under multiple problem instances. Experiments showed that it outperforms many existent crossovers. An interesting conclusion is that Transshipments offer an important flexibility to systems that faces embarrassing storage capacity limits. The observed results confirm the success of evolutionary algorithms in solving inventory problems. Future studies will be concentrated on two directions:

- The multi-objective optimization of multi-location systems with storage capacity, where costs, lead times and service level should be optimized.
- The amelioration of real-coded evolutionary algorithms by incorporating effective search and sensitivity estimation techniques in crossover or mutation operators.

REFERENCES

[1] Arnold, J. and Köchel, P., Evolutionary Optimization of the Multi-location Inventory Model with Lateral Transshipments, 1997.

[2] Herer, Y.T, Tzur, M., and Yücesan, E., The multi-location Transshipment problem. (Forthcoming in *IIE Transactions*), 2005.

[3] Aggarwal, S.P., Inventory control aspect in warehouses. Symposium on Operations Research, Indian National Science Academy, New Delhi. 1967.

[4] Krishnan, K.S. and Rao, V.R.K., Inventory control in N warehouses. J. Industrial. Engineering, Vol.16, No.3, pp. 212-215. 1965.

[5] Jonsson, H. and E.A. Silver. Analysis of a Two-Echelon Inventory Control System With Complete Redistribution. Management Science 33, 215-227. 1987.

[6] Tagaras, G., Effects of pooling on the optimization and service levels of two-location inventory systems. IIE Trans., Vol. 21, No. 3, pp. 250-257. 1989.

[7] Rudi, N., SANDEEP KAPUR, AND DAVID PYKE (1998). A Two–Location Inventory Model with Transhipment and Local Decision Making.

[8] Özdemir, D., E. Yücesan, and Y.T. Herer. 2003. Multi-Location Transshipment Problem with capacitated Transportation. Technology Management Area, INSEAD. *Proceedings of the 2003 Winter Simulation Conference.*

[9] Glasserman P., Gradient estimation via perturbation analysis, Kluwer Academic Publishers, Hingham. 1991.

[10] Herer, Y. and Rashit, A., Lateral Stock Transshipments in a Two-location Inventory System with Fixed Replenishment Costs. Department of Industrial Engineering, Tel Aviv University. 1999a

[11] Herer, Y. and Rashit, A., Policies in a general two-location infinite horizon inventory system with lateral stock Transshipments. Department of Industrial Engineering, Tel Aviv University. 1999b.

[12] H.-G. Beyer. Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. Computer Methods in Applied Mechanics and Engineering, 186(2-4):239267. 2000.

[13] Rechenberg, I., *Evolution Strategy* , in Zuarda et.al. 1994, pp.147-159

[14] Fogel, D. B. (1995). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway, New Jork.

[15] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Anna Arbor.