With Parallel Evolution towards the Optimal Order Policy of a Multi-Location Inventory with Lateral Transshipments

by

Jens Arnold, Peter Köchel, Holger Uhlig
Department of Computer Science
Technical University of Chemnitz
09107 Chemnitz, Germany

Abstract

The problem to find an optimal control for a multi-location inventory model with lateral transshipments in general can not be solved in an analytical way. The main reason for having only a few results on multi-location models with lateral transshipments is the fact that the solution of such models is connected with a lot of analytical and numerical problems. In the case of non-linear cost functions no analytical solutions exist until now, and they will hardly be expected. The present paper deals with the extension of our inventory model to standing expenses for orders ($K>0$). This new assumption was proved to be the reason for changing the form of the optimization problems' objective function. Hence, there are no information about the structure of the optimal order and transport decisions to this day. Therefore, we will describe an optimization method which is based on the combination of simulation and parallel evolutionary algorithms. The objective is to select the best control strategy from a given set of different strategies and to optimize the parameters of this strategy.

Key words: multi-location inventory, evolutionary algorithms, simulation

1. Introduction

The investigation of N-location models with lateral transshipments, $N\geq 2$, is an important problem for mathematical inventory theory as well as for inventory practice. The combination of N locations can result in a system with vertical, horizontal, and mixed structure. While Echelon-models are widely investigated the results on systems with horizontal structure are smaller. The two-location-one period case with linear cost functions is considered by AGGARWAL (1967) and KÖCHEL (1975). KÖCHEL (1982) investigates the N-location-infinite-period model with linear cost functions for the first time. The effect of lateral transshipment on the service levels in a two-location-one-period model is studied in TAGARAS (1989). A dynamic model with finite horizon is examined by ROBINSON (1990). In all works no standing expenses are assumed. Some results on a two-location-one-period model with standing expenses for orders are given in HERER and RASHIT (1995).

Thus corresponding investigations should be concentrated on algorithmic solutions and/or various search procedures. One very promising attempt in this direction is the application of evolutionary algorithms (EA), which was presented in a paper by ARNOLD (1995) at 2nd ISIR-Summer School 1995 in Portorož. In addition to this paper we extended our inventory model to standing expenses for an order ($K>0$). This new assumption was proved to be the reason for changing the form of the optimization problems' objective function. Hence, there are no information about the structure of the optimal order and transport decisions to this day.

The investigated model is described briefly in the next section. Section 3 gives an survey of the simulation algorithm for our multi-location inventory model. Three main order policies are introduced in section 4. A special parallel co-evolution is shown in section 5. The main results and an outlook for further investigations will be outlined in the final section 6.

2. A Multi-Location Inventory Model with Transshipments

The investigated multi-location inventory model with transshipments is described just briefly. For further and more detailed information have a look at the paper of professor KÖCHEL included in these proceedings.

There are $N \geq 2$ locations which are all storing the same product. At the beginning of a period $t=1, 2, \ldots$ additional inventory can be ordered by an order decision (OD). Ordered inventory is received immediately by cost $k_i>0$ (*ordering costs*) for one unit at location i and *standing expenses* for an order $K>0$ for all locations. During period t demand occurs in accordance with a random vector $\underline{\mathbf{s}}(t) = (\underline{s}_1(t), \ldots, \underline{s}_N(t))$. It is assumed that $\{ \underline{\mathbf{s}}(t); t = 1, 2, \ldots \}$ is a sequence of independent identically distributed random vectors. Let $E[\underline{s}_i(t)] =: m_i$ exist with $0 < m_i < \infty$ for $i=1(1)N$. At the end of a period it is possible to redistribute the present stock by a transport decision (TD). Transshipment occurs immediately by cost $c_{ij}>0$ (*transport costs*) for one unit transported from location i to location j. There are also *standing expenses for transport* $C>0$ possible. After this the unsatisfied demand is backlogged and costs are incurred - *holding cost* $h_i>0$ per unit of undemanded inventory or *shortage cost* $p_i>0$ per unit of unsatisfied demand at location i. The problem is to find a policy (i.e. a sequence of OD's and TD's) which minimizes the expected average cost over an infinite horizon. Such a policy is called "average optimal".

Three questions are interesting with reference to the described inventory model:
1. Is there any optimal policy?

If yes:

2. How can we find such a policy?
3. How much mean costs are expected per period?

A positive answer to the first question is given by the results of the so called periodical steady stated MARKOVian decision model (see KÖCHEL 1980, 1982 and 1988). That means the optimal policy is *steady*: For all periods $t=1, 2, \ldots$ the same policy for choosing an OD and a TE is valid.

The other two questions are answered by the following statements of KÖCHEL 1982 and KÖCHEL 1988 about this inventory model with K=0 (no standing expenses for any order):

(i) A solution $\mathbf{S}^* = (S_1, \ldots, S_N)$ of the convex optimization problem $\min\{ g(\mathbf{a}): \mathbf{a} \geq \mathbf{0} \}$ exists with

$$g(\mathbf{a}) = g(a_1, \ldots, a_N) = \sum_{i=1}^{N} [k_i m_i + L_i(a_i)] - C(\mathbf{a}) \tag{1}$$

as the mean value of expected costs at one period with a vector of stocks $\mathbf{a} \geq \mathbf{0}$ and optimal TD. In this connection

$$L_i(a_i) = \mathbf{E}[h_i \max(0; a_i - \underline{s}_i) + p_i \max(0; \underline{s}_i - a_i)]$$

represents the expected holding and shortage costs at location i with stock a_i in one period, $i=1(1)N$. $C(\mathbf{a})$ is the expected reduction in costs by the optimal TD for an vector of stocks $\mathbf{a}$ generated by the OD.

(ii)The optimal OD is of a (S, S)-type (base stock policy). That means, at the beginning of a period the vector of stocks $\mathbf{x} = (x_1, \ldots, x_N) \leq \mathbf{S}^*$ is stocked up on the vector $\mathbf{S}^*$.

(iii)The optimal TD is the solution of an open linear transporting problem with the cost coefficients $C_{ij} := h_i + p_j - c_{ij} + k_j - k_i$, $i, j=1(1)N$, $i \neq j$, with locations of positive remaining stock as suppliers and locations of negative remaining stock as consumers.

Now we turn us towards the case of $K>0$ standing expenses for orders. For this no statements about the form of the optimal OD or TD are proved. That's why we recommend to use an order decision of $(\sigma, \mathbf{S})$-type, shown in figure 1.

It could be described in general: If $\sigma \subseteq \mathbf{R}^N$ and $\mathbf{S} \in \mathbf{R}^N$ are fixed then we will apply to the current stocking vector $\mathbf{x} \leq \mathbf{S}$ the rule

IF $\mathbf{x} \in \sigma$ **THEN**	$\mathbf{x} := \mathbf{S}$	{ restock to $\mathbf{S}$ }
ELSE		{ no order }.

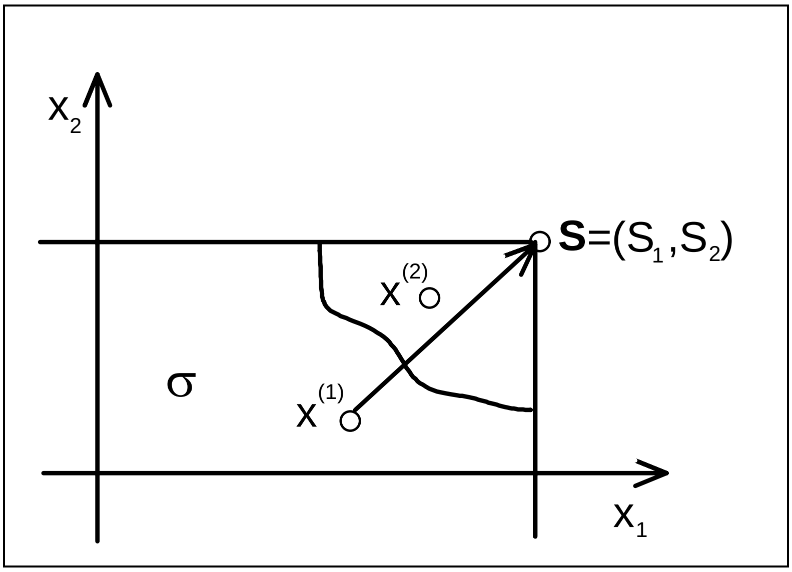

Figure 1: The effect of a (σ, $\mathbf{S}$)-order decision for a two-location inventory for example. The set $\sigma \subseteq \mathbf{R}^N$ is called the *order domain*, the point $\mathbf{S} \in \mathbf{R}^N$ is the *restocking level*.

The recommendation to use an order decision of (σ, $\mathbf{S}$)-type seems to be most practical because this rule is also optimal with $K>0$ for the one-inventory-model (here are $S \in \mathbf{R}^1$ and $\sigma = \{ x \in \mathbf{R}^1 : x < s$ for a $s \in \mathbf{R}^1$ with $s<S \}$, see GIRLICH, KÖCHEL and KÜENLE (1990)) as well as for the multi-goods model (see KÜENLE 1986). To explain it in simple words: Since we order it should pay. That means the fixed order costs K plus the expected costs for the optimal stocking level should be lower than the expected costs for the case of "doing nothing".

After confining us to the OD's class of (σ, $\mathbf{S}$)-type, we search for the best strategy and its optimal parameters σ^* and $\mathbf{S}^*$ now. That could be done in two steps:

Step 1: Determination of the optimal restocking level $\mathbf{S}^*$ as the minimum of function g from equation (1).

Step 2: Determination of the optimal order domain $\sigma^* := \{ \mathbf{x} \in \mathbf{R}^N : K + g(\mathbf{S}^*) < g(\mathbf{x}) \}$.

However, there is generally no analytical solution for the function g with $N>2$ defined with equation (1). The reason is that there is no closed solution for the open linear transporting problem with more than two locations (statement (iii)). Hence, the function $C(.)$ which is used in g and accordingly g itself can't be expressed in numbers. To clear this hurdle we use simulation.

3. Simulation

Obviously the value of $C(\mathbf{a})$ is a mean for every valid vector of stocks $\mathbf{a}$: $C(\mathbf{a}) = \mathbf{E}[\, C(\mathbf{a}, \underline{\mathbf{s}})\,]$ and $g(\mathbf{a}) = \mathbf{E}[\, g(\mathbf{a}, \underline{\mathbf{s}})\,]$. $C(\mathbf{a}, \underline{\mathbf{s}})$ and $g(\mathbf{a}, \underline{\mathbf{s}})$ are two random variables representing the reduction in costs and the value of costs in one period with OD $\mathbf{a}$ and the optimal TD. Let $\mathbf{s}$ be one realization of the demand $\underline{\mathbf{s}}$ and $g(\mathbf{a}, \underline{\mathbf{s}})$ the costs in one period with OD $\mathbf{a}$, demand $\mathbf{s}$

and an optimal TD. Because we can determine the optimal TD for every given vector of stocks $\mathbf{a}$ and every realization $\mathbf{s}$ of demand, the sampling mean

$$g^{(n)}(\mathbf{a}) := 1/n\,[\,g(\mathbf{a}, \mathbf{s}^{(1)}) + ... + g(\mathbf{a}, \mathbf{s}^{(n)})\,] \qquad (2)$$

for every sampling

$$\mathbf{s}^{(1)} = (s_1^{(1)}, ..., s_N^{(1)}),\ \mathbf{s}^{(2)} = (s_1^{(2)}, ..., s_N^{(2)}), ..., \mathbf{s}^{(n)} = (s_1^{(n)}, ..., s_N^{(n)}) \qquad (3)$$

of demand and every sampling rate $n \geq 1$ is an estimated value of $g(\mathbf{a})$ and the estimate function $g^{(n)}$ is a convex function in reference to $\mathbf{a}$ for every n and every sampling (because the convexity of function g) the following simple method for simulation based optimization of g can be used:

1. A sampling of demand (3) with rate n is generated accordingly to the distribution function of $\underline{\mathbf{s}}$ by simulation.
2. The accompanying estimated value $g^{(n)}(\mathbf{a})$ is computed for given ODs **a** in accordance with (2).
3. Any method for the optimization of convex functions is used to determine a better OD.
4. Stop, if no improvement is possible.
5. After abort the OD $\mathbf{a}'$ and the accompanying estimated value $g^{(n)}(\mathbf{a}')$ are considered to be an approximation to the optimal OD $\mathbf{a}^*$ and the minimum expected costs $g(\mathbf{a}^*)$.

Now we can give a brief description of what a simulation (evaluator) has to do:

Restrictions number of locations: N
number of periods: n

1. **Initial values** $x_i = x_0$, $i=1(1)N$ { $\mathbf{x_0}$ optimal stocks for independent locations }

2. **Order decision** applying order policy

order costs:

$$G_{bestell}(k) = \begin{cases} K + \sum_{i=1}^{N} k_i (S_i^* - x_i) & \text{, if } \mathbf{x} \text{ is in order domain} \\ 0 & \text{, else} \end{cases}$$

vector of stocks:

$$a_i = \begin{cases} S_i^* & \text{, if } \mathbf{x} \text{ is in order domain} \\ x_i & \text{, else} \end{cases}$$

3. **Demand** realized according to the probability distribution

vector of demand: s_i = *distribution$_i$* (distribution parameter)
vector of stocks: $y_i = a_i - s_i$

4. **Transport**

transport matrix: $\{ t_{ij} \}$ as optimal TD

transport costs:

$$G_{transp}(k) = \sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij}\, c_{ij}$$

vector of stocks:

$$y'_i = \begin{cases} y_i - \sum_{j=1}^{N} t_{ij} & \text{, if } y_i > 0 \\ y_i + \sum_{j=1}^{N} t_{ji} & \text{, if } y_i < 0 \end{cases}$$

5. **Holding**

holding costs: $G_{lager}(k) = \sum_{i=1}^{N} [\, h_i \max(0, y'_i) + p_i \max(0, -y'_i) \,]$

6. **Repeat** 2. to 5. n times

7. **Output** estimation

$$\hat{g}(\text{ order domain}, \mathbf{S}^*) = 1/n \sum_{k=1}^{n} [G_{bestell}(k) + G_{transp}(k) + G_{lager}(k)].$$

4. Determination of the Optimal Order Domain

As described in section 2 the determination of the optimal order domain σ^* is another partial problem of our optimization. If we assume a continual demand then we will have to decide the affiliation to the order domain σ^* for an infinity set of stock vectors. At the moment we choose an approximate solution according to the following statement:
The set

$$\overline{\sigma}^* := \{ \mathbf{x} \in \mathbf{R}^N : \mathbf{x} \leq \mathbf{S}^* \} \setminus \sigma^*$$

is convex which follows from the convexity of function g. The approximation consists now of the restriction to a set of simple structured classes of $\overline{\sigma}$ -sets (and so σ-sets). We consider the following classes (called “strategies“ in future) in this paper: “rectangle“-sets, “triangle“-sets and “ellipse“-sets.

- <u>Rectangle-Strategy:</u>

The order domain σ is described by

$$\sigma(\mathbf{s}) := \{ \mathbf{x} = (x_1, x_2, ..., x_N) : x_1 \leq s_1 \vee x_2 \leq s_2 \vee ... \vee x_N \leq s_N \}$$

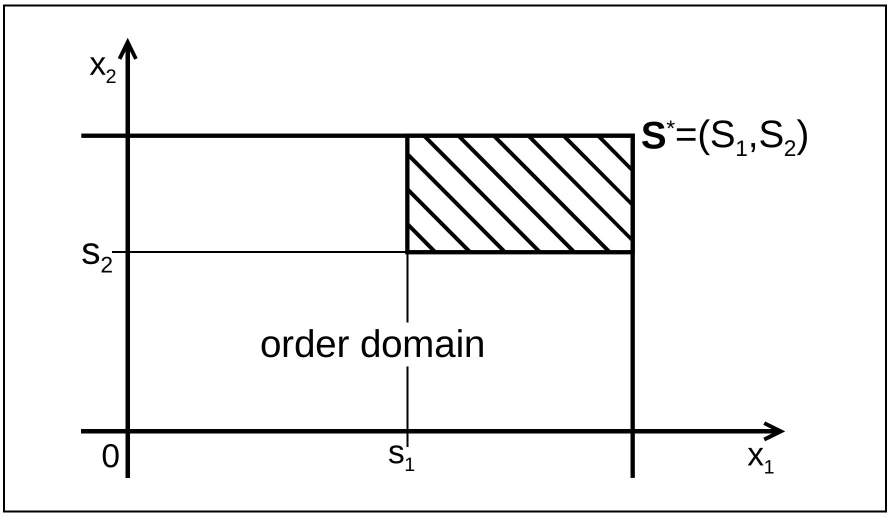

Figure 2: The rectangle-strategy for 2 locations with $\mathbf{s} = (s_1, s_2, ..., s_N) \leq \mathbf{S}^*$.

- <u>Triangle-Strategy:</u>

The order domain σ is described by a hyperplane through N points. The coefficients for the equation of the plane are obtained by the solution of a N+1 dimensional linear equation system.

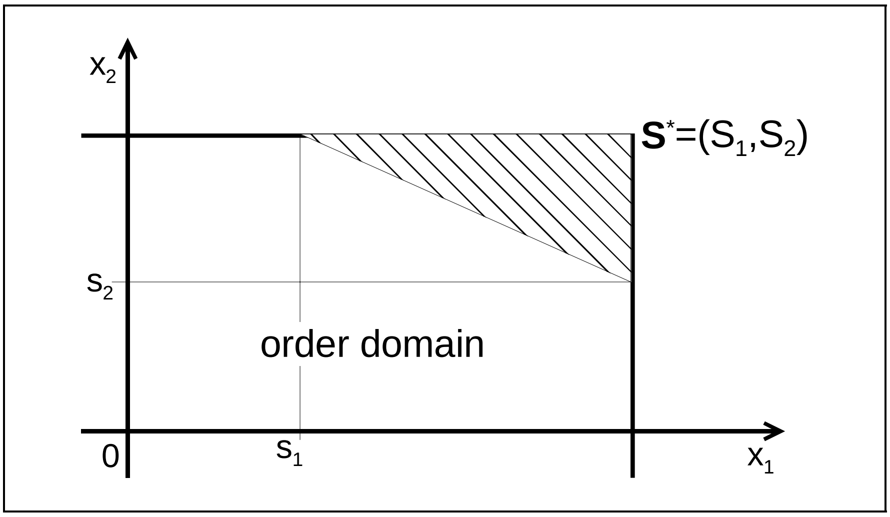

Figure 3: The triangle-strategy for 2 locations with $\mathbf{s} = (s_1, s_2, ..., s_N) \leq \mathbf{S}^*$.

- <u>Ellipse-Strategy:</u>

The order domain σ is described by a hyperellipse through N points. The coefficients for the equation of the ellipse are obtained by the semiaxes from the centre $\mathbf{S}^*$ by $1/(S_i-s_i)$, $i=1(1)N$.

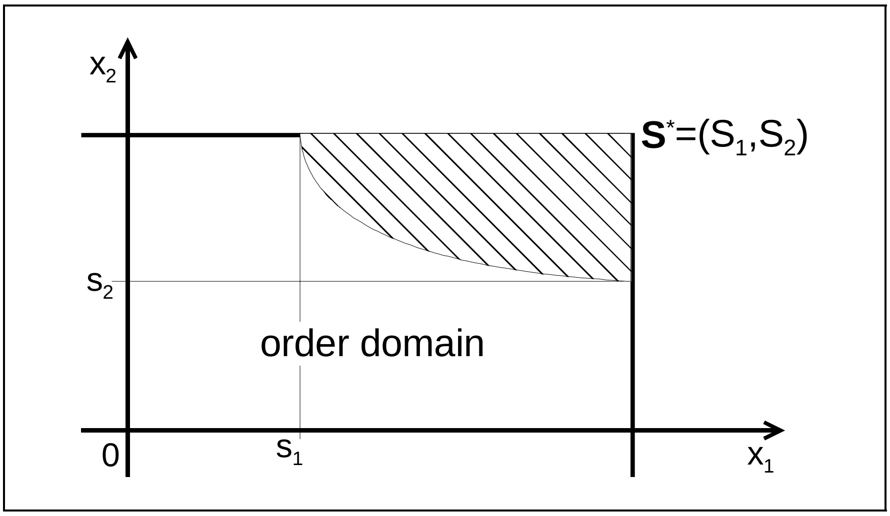

Figure 4: The ellipse-strategy for 2 locations with $\mathbf{s} = (s_1, s_2, ..., s_N) \leq \mathbf{S}^*$.

Now it is quite clear that we have to search for both a *function* which allows a good approximation to the bounds of σ (see figures 2 to 4) and the stocking levels $\mathbf{s}^*$ and $\mathbf{S}^*$ (as *parameters* of this function) for an optimal σ^*. In the next section we discuss a parallel co-evolution of different populations (each for a special strategy) distributed on several processors. Our aim is to find the best strategy and its optimal parameters $\mathbf{s}^*$ and $\mathbf{S}^*$.

5. Optimization with Parallel Co-Evolution

From the mathematician's, engineer's and computer scientist's point of view the evolution is an extreme powerful optimization method. Basic notions and concepts of evolutionary optimization, as well as a long list of references was given in a paper by ARNOLD (1995), presented at 2nd ISIR-Summer School 1995 in Portorož. You can find a good introduction to this topic in MICHALEWICZ (1992). Throughout the paper we will use biological terms. We believe that this is a source of inspiration and helps one to understand the algorithm intuitively. In many papers, it has been reported that evolutionary algorithms can find better solutions than heuristic algorithms for many problems. However, evolutionary algorithms require a lot of computation time for simulating the survival of the fittest of a number of individuals. Therefore, in order to outperform heuristic algorithms in both the quality of solutions and the search time, we need parallel processing of co-evolution.

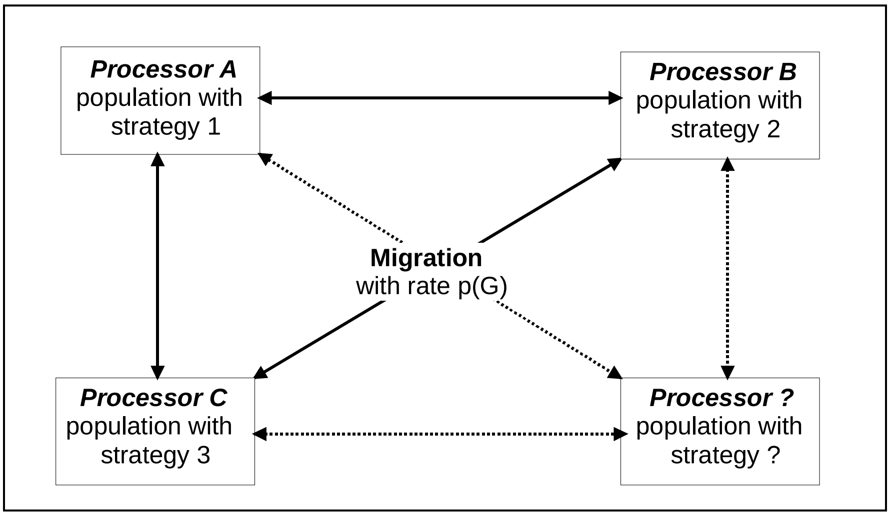

Figure 5: The concept of parallel co-evolution with 3 different strategies. At the beginning each population has a constant number of individuals which all use the same strategy.

Figure 5 shows our concept of parallel co-evolution. It is implemented with Parallel Virtual Machine (PVM), version 3. We use one PVM-process(or) for each strategy. At the beginning each population i has a constant number of individuals P_i which all use the same strategy. The individuals are represented by two chromosomes: a vector $\mathbf{s}^*$ and a vector $\mathbf{S}^*$, each of real numbers. After a number t of generations with isolated life (populations of islanders) some individuals (independent of their fitness!) spread out and migrate to other populations by random. These individuals will deleted in their origin population. That means, the number of individuals in each isolated population can increase and decrease dynamically during the optimization process, but the total of all individuals is constant. Now the isolated life goes on, but t should be decreased after each migration cycle for a better performance.

This “coarse-grained parallel evolution algorithm“, in which several large populations are evolved in parallel with little interaction, was described quite often in literature (BELEW and BOOKER (1991), FORREST (1993), SCHWEFEL and MAENNER (1991)). The communication overhead is reduced to a small number of floating individuals. Each processor only sends out some individuals to some other processors after t generations. No other information is exchanged between processors.

At the moment we suggest the following parameters for the evolution

- $P_i = 10$ at $t = 0$ for all strategies i;
- $t = 200$ for the first migration cycle;
- migration rate p(G) =0.5

These parameters (especially P_i) should be adjusted to the specific computer hardware. The evolution described above does the following: At first each processor optimizes the parameters of *one* strategy by reproduction, recombination, mutation and selection. This job is done by an evolution strategy that takes place inside of each processor (see figure 6). Notice, mating is always possible only with two individuals of the same strategy! After this we have a set of isolated populations with individuals of a relatively high fitness (minimal inventory costs).

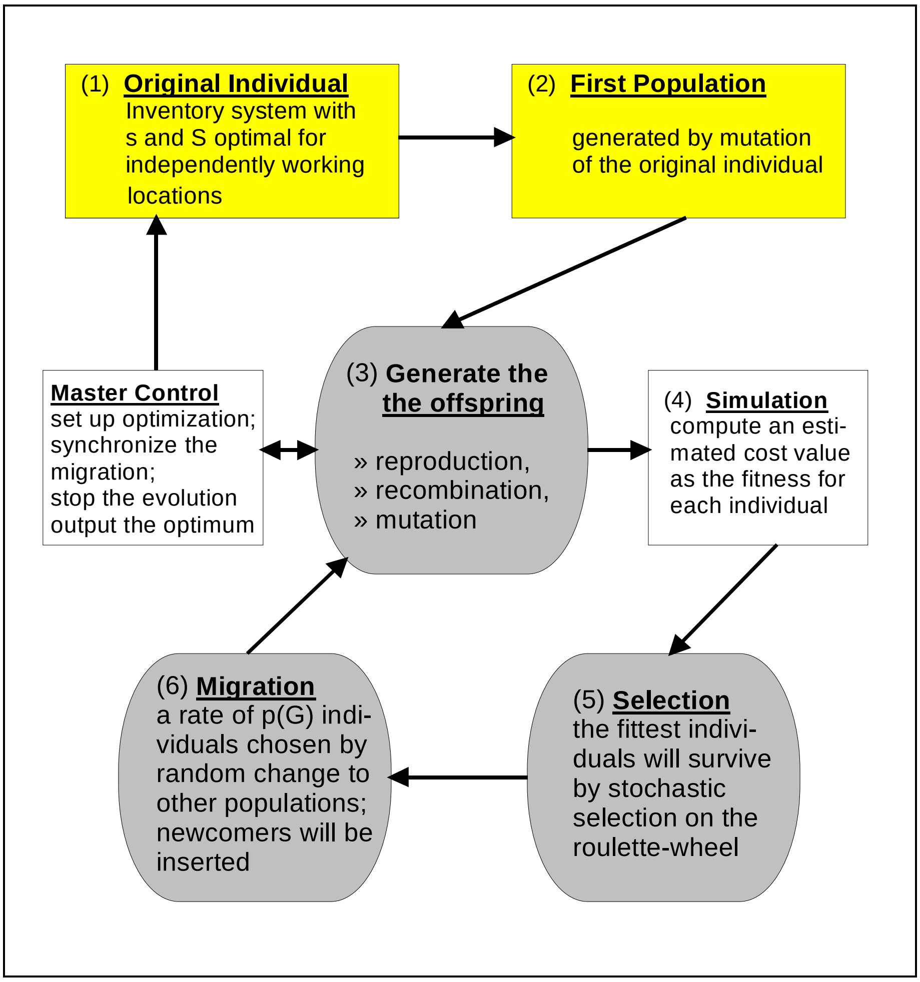

Figure 6: The evolution inside each isolated population

This is necessary to make sure that the individuals which now migrate and start a direct ranking between different strategies have nearly optimal parameters for their own strategy. To the end of the evolution only one strategy should survive. The individuals of that strategy will live in all populations and the whole performance of the distributed hardware is used to find the optimal parameters $\mathbf{s}^*$ and $\mathbf{S}^*$ for the best strategy. You can see that our method combines the two optimization steps of the conclusions carried out at the end of section two.

The optimization should be aborted if there is no improvement over a *lot of* generations. A “lot of“ means that we cannot give any more accurate value (see ARNOLD (1995)).

6. What is there to be done?

In this paper, we have proposed a new parallel evolutionary algorithm which gives the possibility to optimize both strategy and parameters of the strategy for the order policy of a multi-location inventory with transshipments. Unfortunately, we cannot present any exemplary numerical results so far. The reason for this lack of information is that we have to improve our simulator. Some problems came up only after a few of test runs: The open linear transporting problem need more investigation to consider the standing expenses for ordering (K>0). The Gaussian elimination method for solving the system of linear equations of a plan equation of the hyperplane for the “triangle“-strategy had to be redesigned just like that. The “ellipse“-strategy is a new extension of the unsuitable “circle“-strategy. Nevertheless, you will find some interesting results for the “triangle“-strategy in the paper of HADER included in these proceedings.

References

Aggarwal, S.P. (1967)
Inventory control aspect in warehouses. Symposium on Operations Research, INDIAN NAT: SCI: ACADEMY, New Dehli

Arnold, J. (1995)
Evolutionary Algorithms for Optimization of Badly Structured or Unknown Functions Like Inventory Costs for Example. Lecture Notes of the ISIR Int. Postgraduate Summer School, Vol. 2, Budapest, Portorož

Belew, R.K. and Booker, L.B. (Ed.) (1991)
Proceedings of the fourth international conference on Genetic Algorithms. Morgan Kaufmann Publishers, Inc., San Mateo

Forrest, S. (Ed.) (1993)
Proceedings of the fifth international conference on Genetic Algorithms. Morgan Kaufmann Publishers, Inc., San Mateo

Girlich, H.J.; Köchel, P. and Küenle, H.U. (1990)
Steuerung dynamischer Systeme. Fachbuchverlag, Leipzig

Herer, Y. and Rashit, A. (1995)
Lateral stock transshipments in a two location inventory system. Proceedings of the 13th Intern. Conference on Production Research, Jerusalem, Israel, pp. 593-595

Köchel, P. (1975)
Ein stochastisches Lagerhaltungsmodell für mehrere miteinander verbundene Lager. Math. Operationsforschung und Statistik, vol. 6, No. 3, pp. 413-426

Köchel, P. (1980)
Periodisch stationäre Markovsche Entscheidungsmodelle. Elektron. Informationsverar-beitung u. Kybernetik 16, pp 553-567

Köchel, P. (1982)
Ein dynamisches Mehr-Lager-Modell mit Transportbeziehungen zwischen den Lagern. Math. Operationsforsch. und Statistik, Ser. Optimization, vol. 13, No. 2, pp. 267-286

Köchel, P. (1988)
Optimal adaptive inventory control for a multi-location model with redistribution. Optimization 18, pp 525-537

Küenle, H.U. (1986)
Stochastische Spiele und Entscheidungsmodelle. Teubner Texte zur Mathematik, Volume 89, Leipzig

Michalewicz, Z. (1992)
Genetic Algorithms + Data Structures = Evolution Programs. Published by Springer, Berlin, Heidelberg

Robinson, L.W. (1990)
Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. Operations Research, vol. 38, No. 2, pp. 278-295

Schwefel, H. and Maenner, R. (Ed.) (1991)
Parallel Problem Solving from Nature. Published by Springer, Berlin, Heidelberg

Tagaras, G. (1989)
Effects of pooling on the optimization and service levels of two-location inventory systems. IIE Trans., vol. 21, No. 3, pp. 250-257

Uhlig, H. (1996)
Simulation and Evolutionary Algorithms: A way to optimal control of models with distributed resources. Diploma Thesis at TU Chemnitz-Zwickau, Dep. of Computer Science