2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Supply Chain Optimization Using Chaotic Differential Evolution Method

Leandro dos Santos Coelho and Heitor Silvério Lopes

Abstract— **This paper describes the application of differential evolution approaches to the optimization of a supply chain. Although simplified, this supply chain included stocks, production, transportation and distribution, in an integrated production-inventory-distribution system. The supply chain problem model is presented as well as a short introduction to each evolutionary algorithm. Differential evolution (DE) is an emergent evolutionary algorithm that offers three major advantages: it finds the global minimum regardless of the initial parameter values, it involves fast convergence, and it uses few control parameters. Inspired by the chaos theory, this work presents a new global optimization algorithm based on different DE approaches combined with chaotic sequences (DEC), called chaotic differential evolution algorithm. The performance of three evolutionary algorithm approaches (genetic algorithm, DE and DEC) and branch and bound method were evaluated with numerical simulations. Results were also compared with other similar approach in the literature. DEC was the algorithm that led to better results, outperforming previously published solutions. The simplicity and robustness of evolutionary algorithms in general, and the efficiency of DEC, in particular, suggest their great utility for the supply chain optimization problem, as well as other logistics-related problems.**

I. Introduction

THE optimization of a supply chain is an integer programming problem or a constrained integer-mixed problem [1]. Depending on how it was formulated, it can be a very hard problem for classical optimization methods. Consequently, several methodologies for optimizing a supply chain have been proposed in the literature so far. These methodologies can be organized into four main categories: (i) stochastic approximation or gradient-based methods; (ii) meta-models, such as response surface, artificial neural networks and fuzzy systems; and (iii) random search-based methods. Regarding evolutionary algorithms, genetic algorithms are the most popular for supply chain optimization problems. See, for instance, Hwang [2], Lee *et al.* [3], Syarif *et al.* [4], Zhou *et al.* [5] and Smirnov *et al.* [6].

The objective of this work is to compare evolutionary algorithms (EAs) for the optimization of a supply chain, based on a benchmark case study proposed by Mak and Wong [1]. EAs use a population of structures (individuals) which, in turn, represent points in the search space of possible solutions to a given problem. The following EAs are evaluated and compared: (i) genetic algorithm [1], (ii) differential evolution (DE), and (iii) new differential evolution approaches based on chaotic sequences (DEC).

The rest of the paper is organized as follows: section 2 describes the methodology and the scope of problem of supply chain's optimization, while section 3 explains the genetic algorithm, standard DE and the DEC. Section 4 presents the results of the supply chain's optimization and compares methods to solve the case study. Lastly, section 5 outlines our conclusions and future research.

II. Methodology

A. Scope of Problem

The supply chain analyzed in this work was based in the model proposed by Mak and Wong [1]. A simplified block diagram of this supply chain is presented in Fig. 1.

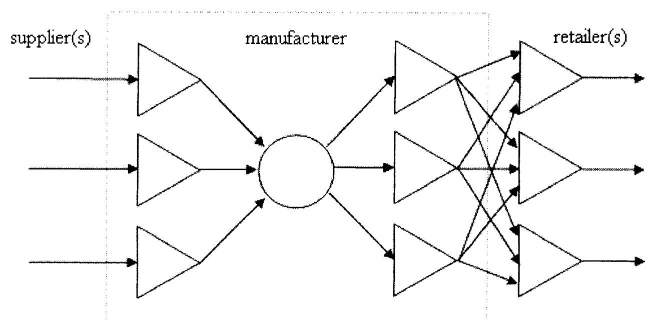

Fig 1. Block diagram of the supply chain that integrates production, stocking and distribution systems.

The diagram of Figure 1 consists of three different sectors serially arranged, and includes suppliers, a manufacturer and retailers. Suppliers deliver raw materials to manufacturers, who, in turn, produce goods. Both raw materials and final products are stored in manufacturer's warehouses. Products are further transported to retailers in different regions. The mathematical model that describes such system can be

Manuscript received March 12, 2006.

L. S. Coelho is with the Pontifical Catholic University of Parana, Production and Systems Engineering Graduate Program, Automation and Systems Laboratory, PUCPR/PPGEPS/LAS, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Parana, Brazil (e-mail: leandro.coelho@pucpr.br).

H. S. Lopes is with Federal Technological University of Parana, UTFPR/CPGEI, Av. 7 de setembro, 3165 – 80230-901, Curitiba, Parana, Brazil (e-mail: hslopes@cpgei.cefetpr.br).

conceived in a simplified manner, for the purpose of comparing the heuristic optimization algorithms. The model is evaluated by an objective function to be minimized subject to a set of constraints. The object function shown in Equation 1 comprises costs of storage, manufacturing, transportation and shortage of products [1]:

$$eval(x) = C_{storage} + C_{manufact} + C_{transport} + C_{shortage} \quad (1)$$

such that:

$$C_{storage} = \sum_{r=1}^{R}\sum_{p=1}^{P}\sum_{t=2}^{T+1} H_{rp}^{R} K_{rpt} + \sum_{p=1}^{P}\sum_{t=2}^{T+1} H_{p}^{P} J_{pt} + \sum_{m=1}^{M}\sum_{t=2}^{T+1} H_{m}^{M} I_{mt} \quad (2)$$

$$C_{manufact} = \sum_{p=1}^{P}\sum_{t=1}^{T} C_{p}^{P}\left[J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{pt}\right] \quad (3)$$

$$C_{transport} = \sum_{r=1}^{R}\sum_{p=1}^{P}\sum_{t=1}^{T} C_{rp}^{D} Z_{rpt} + \sum_{m=1}^{M}\sum_{t=1}^{T} C_{m}^{M}\left\{I_{m,t+1} + \sum_{p=1}^{P}\theta_{mp}\left[J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{pt}\right] - I_{mt}\right\} \quad (4)$$

$$C_{shortage} = \sum_{r=1}^{R}\sum_{p=1}^{P}\sum_{t=1}^{T} C_{rp}^{S}\left[D_{rpt} - \left(K_{rpt} + Z_{rpt} - K_{rp,t+1}\right)\right] \quad (5)$$

subject to the following constraints:

$$K_{rpt} + Z_{rpt} - K_{rp,t+1} \geq 0 \quad (6)$$

$$K_{rpt} + Z_{rpt} - K_{rp,t+1} \leq D_{rpt} \quad (7)$$

$$J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{pt} \geq 0 \quad (8)$$

$$\sum_{p=1}^{P} B_{p}\left[J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{pt}\right] \leq \beta_{t} \quad (9)$$

$$\sum_{r=1}^{R}\sum_{p=1}^{P} W_{p}^{P} Z_{rpt} \leq \omega_{t}^{P} \quad (10)$$

$$I_{m,t+1} + \sum_{p=1}^{P}\theta_{mp}\left[J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{pt}\right] - I_{mt} \geq 0 \quad (11)$$

$$\sum_{m=1}^{M} W_{m}^{M}\{I_{m,t+1} + \sum_{p=1}^{P}\theta_{mp}\left[J_{p,t+1} + \sum_{r=1}^{R} Z_{rpt} - J_{p,t}\right\} - I_{mt} \leq \omega_{t}^{M} \quad (12)$$

where:

- B_p : is the process time necessary to manufacture each unity of the p-th product;
- β_t : is the total capable time for manufacturing at the the t-th period;
- C_{rp}^{D} : is the cost of delivering one unity of the p-th product from the manufacturer to the r-th retailer;
- C_{m}^{M} : is the cost of delivering one unity of the m-th raw material from the supplier to the manufacturer;
- C_{p}^{P} : is the cost of manufacturing each unity of the p-th product;
- C_{rp}^{S} : is the cost of shortage of each unity of the p-th product from the manufacturer to the r-th retailer;
- D_{rpt} : is the demand of the p-th product from the manufacturer to the r-th retailer in the t-th period;
- H_{m}^{M} : is the storage cost for each unity of the m-th raw material kept in the inlet stock of the manufacturer;
- H_{p}^{P} : is the storage cost of each unity of the p-th product kept in the outlet stock of the manufacturer;
- H_{rp}^{R} : is the storage cost of each unity of the p-th product kept in the r-th retailer;
- I_{mt} : is the amount of the m-th raw material stored kept in the inlet stock of the manufacturer, at the beginning of the t-th period;
- J_{pt} : is the amount of the p-th product stored in the manufacturing sector, at the beginning of the t-th period;
- K_{rpt} : is the amount of the p-th product stored in the r-th retailer, at the beginning of the t-th period;
- W_{m}^{M} : is the weight of each unity of the m-th raw material;
- W_{p}^{P} : is the weight of each unity of the p-th product;
- ω_{t}^{M} : is the load limit for transporting materials from supplier to manufacturer at the t-th period;
- ω_{t}^{P} : is the load limit for transporting products fom manufacturer to retailers at the t-th period;
- Z_{rpt}^{M} : is the amount of the p-th product sent from the manufacturer to the r-th retailer, at the t-th period;
- θ_{mp}: is the amount of the m-th raw material necessary to produce each unity of the p-th product.

The objective function (Equation 1) minimizes the sum of the costs relative to storage, manufacture, transport, and product shortage. Equations (2), (3), (4), and (5) define the composition of the costs relative to storage ($C_{storage}$), manufacture ($C_{manufact}$), transport ($C_{transport}$), and product shortage ($C_{shortage}$), respectively. Equations (6) and (8) impose that both sales and production must be positive. In the same way, the inequation (11) imposes that the amount of raw material sent from suppliers to manufacturer must be also positive. Equation (7) limits sales up to the demand of the product, for each period and retailer. Equation (9) limits the production capacity to a given value. Equations (10) and (12) limit, respectively, the total weight of the transported products and raw materials.

The approach adopted for this case study was formulated like an integer programming problem, in which the decision variables that compose vector x, to be optimized by the DE methods, are:

$$\begin{cases} I_{mt} \ (m = 1,2,\dots, M; t = 2,3,\dots, T) \\ J_{pt} \ (p = 1,2,\dots, P; t = 2,3,\dots, T) \\ K_{rpt} \ (r = 1,2,\dots, R; p = 1,2,3,\dots, P; t = 2,3,\dots, T) \\ Z_{rpt} \ (r = 1,2,\dots, R; p = 1,2,3,\dots, P; t = 2,3,\dots, T) \end{cases}$$

where $I_{mt}, J_{pt}, K_{rpt}, Z_{rpt} \geq 0$.

B. Constraint Handling

When applying EAs to the optimization of a supply chain, a key issue is how constraints related to the problem are handled by algorithm. During the last decades, several methods have been proposed for constraint handling in EAs, and they can be grouped into four categories: methods that preserve solutions feasibility, penalty-based methods, methods that clearly distinguish between feasible and unfeasible solutions and hybrid methods.

When EAs are used for constrained optimization problems, it is usual to handle constraints using the concept of penalty functions (that penalize unfeasible solutions). That is, it is tried to solve an unconstrained problem in the search space S using a modified fitness function such as:

$$eval(x) = \begin{cases} f(x), & \text{if } x \in F \\ f(x) + penalty(x), & \text{otherwise} \end{cases} \quad (13)$$

where $penalty(x)$ is zero if no constraint is violated, and it is positive otherwise. Usually, the penalty function is based on a distance measure to the nearest solution in the feasible region F or on the effort to repair the solution. Therefore, Equation (14) shows how the fitness function is primarily defined as a maximization problem, such that

$$fitness = \frac{1}{1 + eval(x)} \quad (14)$$

where x is the set of decision variables to the supply chain problem, that is, $I_{mt}, J_{pt}, K_{rpt}, Z_{rpt}$.

The methodology proposed for constraint handling is divided in two steps. The first step aims at finding solutions for the decision variables that lie within user-defined upper (lim_{upper}) and lower (lim_{lower}) bounds, that is, $x \in [lim_{lower}, lim_{upper}]$. Whenever a lower bound or an upper bound restriction is not satisfied, a repair rule is applied, according to Equations 15 and 16, respectively:

$$x_i = x_i + w \cdot rand[0,1] \cdot \{lim_{upper}(x_i) - lim_{lower}(x_i)\} \quad (15)$$

$$x_i = x_i - w \cdot rand[0,1] \cdot \{lim_{upper}(x_i) - lim_{lower}(x_i)\} \quad (16)$$

where $w \in [0,1]$ is a user-defined parameter and $rand[0,1]$ is an uniformly distributed random value between 0 and 1.

In the second step decision variables are considered inequalities ($g_i(x) \leq 0$). In this work we maximize the fitness function defined in Equation 14, and thus Equation 13 is rewritten as:

$$eval(x) = \begin{cases} f(x), & \text{when } g_i(x) \leq 0 \\ f(x) + r \cdot q \cdot \sum_{i=1}^{r} g_i(x), & \text{when } g_i(x) > 0 \end{cases} \quad (17)$$

where q is a positive constant (arbitrarily set to 500,000) and r is the number of constraints $g_i(x)$ that were not satisfied.

III. Differential Evolution

A. Classical Differential Evolution

Storn and Price [7] first introduced the DE algorithm a few years ago. In 1997, the DE was successfully applied by Storn [8] to the optimization of some well-known non-linear, non-differentiable and non-convex functions. DE is an approach for the treatment of real-valued optimization problems. DE combines simple arithmetic operators with the classical operators of crossover, mutation and selection to evolve form a randomly generated starting p opulation to a final solution.

There are two variants of DE that have been reported, DE/*rand*/1/*bin* and DE/*best*/2/*bin*. The different variants are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromo some that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the crossover mechanism used to create the child population. The *bin* acronym indicates that crossover is controlled by a series of independent binomial experiments.

DE, at each time step, mutates vectors by adding weighted, random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by trial vector in the next generation. The variant implemented in this paper was the DE/*rand*/1/*bin* and it is given by the following steps:

(i) Initialize a population of individuals (solution vectors) with random values generated according to a uniform probability distribution in the n dimensional problem space.

(ii) For each individual, evaluate its fitness value.

(iii) Mutate individuals in according to equation:

$$z_i(t+1) = x_{best}(t) + f_m(t)[x_{i,r_2}(t) - x_{i,r_3}(t)] \quad (18)$$

(iv) Following the mutation operation, crossover is applied in the population. For each mutant vector, $z_i(t+1)$, an index

$rnbr(i) \in \{1, 2, \cdots, n\}$ is randomly chosen, and a *trial vector*, $u_i(t+1) = \left[u_{i1}(t+1), u_{i2}(t+1), \ldots, u_{in}(t+1)\right]^T$, is generated with

$$u_{ij}(t+1) = \begin{cases} z_{ij}(t+1), \\ \text{if (randb(j)} \leq CR) \text{ or } (j = rnbr(i)), \\ x_{ij}(t), \\ \text{if (randb(j)} > CR) \text{ or } (j \neq rnbr(i)) \end{cases} \quad (19)$$

To decide whether or not the vector $u_i(t+1)$ should be a member of the population comprising the next generation, it is compared to the corresponding vector $x_i(t)$. Thus, if F_c denotes the objective function under minimization, then

$$x_i(t+1) = \begin{cases} u_i(t+1), \text{ if } F_c(t+1) < F_c(x_i(t)), \\ x_i(t), \text{ otherwise} \end{cases} \quad (20)$$

(iv) Loop to step (ii) until a stopping criterion is met, usually a maximum number of iterations (generations).

In the above equations, $i = 1,2,\ldots,N$ is the individual's index of population; $j = 1,2,\ldots, n$ is the position in n dimensional individual; t is the time (generation); $x_i(t) = \left[x_{i1}(t), x_{i2}(t), \ldots, x_{in}(t)\right]^T$ stands for the position of the i-th individual of population of N real-valued n-dimensional vectors; $z_i(t) = \left[z_{i1}(t), z_{i2}(t), \ldots, z_{in}(t)\right]^T$ stands for the position of the i-th individual of a *mutant vector*; r_1, r_2 and r_3 are mutually different integers and also different from the running index, i, randomly selected with uniform distribution from the set $\{1, 2, \cdots, i-1, i+1, \cdots, N\}$; $f_m(t) > 0$ is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals so as to avoid search stagnation and it is usually taken form the range [0.1, 1]; *randb(j)* is the j-th evaluation of a uniform random number generation with [0, 1]; CR is a *crossover rate* in the range [0, 1]; and F_c is the evaluation of cost function. Usually, the performance of a DE algorithm depends on three variables: the population size N, the mutation factor f_m, and the crossover rate CR.

B. Chaotic differential evolution

Chaos theory is recognized as very useful in many engineering applications. Chaos is a phenomenon that can appear in solutions for nonlinear differential equations. An essential feature of chaotic systems is that small changes in the parameters or the starting values for the data lead to vastly different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity. These behaviors can be analyzed based on Lyapunov exponents and the attractor theory [9].

Optimization algorithms based on the chaos theory are stochastic search methodologies that differ from any of the existing EAs. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic ergodic searches that depend on probabilities [10].

In DE design, the concepts of optimization based on chaotic sequences can be a good alternative to provide diversity in populations of DE approaches. The parameters f_m, CR and f_m of DE are generally the key factors that affect the DE's convergence. In fact, however, parameters f_m and CR cannot entirely ensure the ergodicity of the optimization in phase search because they are constant factors in classical DE algorithm procedures. Therefore, this paper provides three new approaches introducing chaotic mapping with ergodicity, irregularity and the stochastic property in DE to improve the global convergence. The use of chaotic sequences in EAs can be helpful to escape more easily from local minima than the traditional EAs [10].

One of the simplest dynamic systems evidencing chaotic behavior is the iterator named logistic map [11], whose equation is given by:

$$y(t) = \mu \cdot y(t-1) \cdot \left[1 - y(t-1)\right] \quad (21)$$

where t is the sample, μ is a control parameter, and $0 \leq \mu \leq 4$. The behavior of the system of equation (4) is greatly changed with the variation of μ. The value of μ determines whether y stabilizes at a constant size, oscillates among a limited sequence of sizes, or whether y behaves chaotically in an unpredictable pattern. A very small difference in the initial value of y causes large differences in its long-time behavior (Liu *et al.*, 2005). Equation (4) is deterministic, exhibiting chaotic dynamics when $\mu = 4$ and $y(1) \notin \{0, 0.25, 0.50, 0.75, 1\}$. In this case, $y(t)$ is distributed in the range (0,1) provided that the initial $y(1) \in (0,1)$ and that $y(1) \notin \{0, 0.25, 0.50, 0.75, 1\}$. In this work, $y(1)$=0.48 was adopted for the experiments.

The design of methods to improve the convergence of DE is a challenging issue in EAs. New DE approaches are proposed here. These two new approaches of DE combined with chaotic sequences, DEC, based on logistic maps are described as follows:

Approach 1 – DEC1: The parameter f_m of equation (18) is incremented with the evolution of generations. The value of f_m is modified by the formula (21), based on the following equations:

$$z_i(t+1) = x_{i,r_1}(t) + f_2(t)[x_{i,r_2}(t) - x_{i,r_3}(t)] \quad (22)$$

$$f_1(t) = \left|\mu \cdot f_1(t-1) \cdot \left[1 - f_1(t-1)\right]\right| \quad (23)$$

$$f_2(t) = \left[\left(f_{2f} - f_{2i}\right)\frac{G}{G_{\max}} + f_{2i}\right] f_1(t) \quad (24)$$

where $|\cdot|$ is the absolute value of the expression, f_{2i} and f_{2f} are constants ($f_{2i} < f_{2f}$), and G (the value of G is equal to t) is the current generation number.

Approach 2 – DEC2: The value of the parameter f_m of equation (18) decreases with the evolution of the generations. The value of f_m is modified by same equations as those of DEC1, but $f_{2i} > f_{2f}$.

IV. Simulation Results

The optimization was based on the following assumptions: all stocks (raw materials and products) are initially empty and there are M=3 raw materials, P=2 products, R=3 retailers and T=3 periods. The same parameters of this simplified supply chain problem referred by Mak and Wong[1] were optimized in this work, as follows:

- products demands D_{rpt} at each period are forecasted as:
  $D_{111} = 80; D_{112} = 60; D_{113} = 70;$
  $D_{121} = 50; D_{122} = 50; D_{123} = 55;$
  $D_{211} = 60; D_{212} = 75; D_{213} = 65;$
  $D_{221} = 45; D_{222} = 65; D_{223} = 85;$
  $D_{311} = 80; D_{312} = 70; D_{313} = 90;$
  $D_{321} = 50; D_{322} = 70; D_{323} = 40.$
- Machine processing time, B_p: $(B_1, B_2) = (1, 1)$
- Allotted time for manufacturing, $\beta_t : (\beta_1, \beta_2, \beta_3) = (800,\ 800,\ 800)$
- Transportation cost from manufacturer to retailers, C_{rp}^D: $\left(C_{11}^D, C_{12}^D, C_{21}^D, C_{22}^D, C_{31}^D, C_{32}^D\right) = (1,\ 1,\ 4,\ 4,\ 2,\ 2)$
- Transportation cost from supplier to manufacturer, C_m^M: $\left(C_1^M, C_2^M, C_3^M\right) = (0{,}3;\ 0{,}3;\ 0{,}2)$
- Manufacture cost, C_p^P: $\left(C_1^P, C_2^P\right) = (20, 15)$
- Shortage cost, C_{rp}^S: $\left(C_{11}^S, C_{12}^S, C_{21}^S, C_{22}^S, C_{31}^S, C_{32}^S\right) = (1000,\ 500,\ 1800,\ 1000,\ 1000,\ 1000)$
- Storage cost in the inlet stock, H_m^M: $\left(H_1^M, H_2^M, H_3^M\right) = (5,\ 8,\ 6)$
- Storage cost in the outlet stock, H_p^P: $\left(H_1^P, H_2^P\right) = (4,\ 3)$
- Storage cost of products in the retailers, H_{rp}^R: $\left(H_{11}^R, H_{12}^R, H_{21}^R, H_{22}^R, H_{31}^R, H_{32}^R\right) = (8,\ 4,\ 12,\ 8,\ 8,\ 8)$
- Raw material weight, W_m^M: $\left(W_1^M, W_2^M, W_3^M\right) = (3, 2, 2)$
- Product weight, W_p^P: $\left(W_1^P, W_2^P\right) = (7,\ 13)$
- Load limit from supplier to manufacturer, ω_t^M: $\left(\omega_1^M, \omega_2^M, \omega_3^M\right) = (5000,\ 5000,\ 5000)$
- Load limit from manufacturer to retailers, ω_t^P: $\left(\omega_1^P, \omega_2^P, \omega_3^P\right) = (3000,\ 3000,\ 3000)$
- Amount of raw material used in products, θ_{mp}: $\left(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}\right) = (1,\ 3,\ 2,\ 1,\ 1,\ 2).$

For each of the previously described DE and DEC, a total of 50 experiments were done, using the parameters before mentioned and different initial random seeds. For all optimization algorithms, individuals are composed by the decision variables $I_{mt}, J_{pt}, K_{rpt}, Z_{rpt}$, which are rounded to the nearest integer, when computing the function *eval*(*x*). Variables were allowed to span within the following ranges: $0 \le I_{mt} \le 20$, $0 \le J_{pt} \le 20$, $0 \le K_{rpt} \le 30$ and $0 \le Z_{rpt} \le 120$.

A total of 150,000 fitness evaluations (30 individuals; 5,000 generations) was done by each DE and DEC methods, every run. Other particular parameters used in the standard DE are fixed empirically were:

- DE(1): DE/*rand*/1/*bin* with CR = 0.80 and a constant mutation factor given by $f_m(t) = 0.40$;
- DE(2): DE/*rand*/1/*bin* with CR = 0.80 and a mutation factor $f_m(t)$ given by an uniformly distributed random value between 0.50 and 1.50;
- DE(3): classical DE/*rand*/1/*bin* with CR = 0.80 and the value of mutation factor $f_m(t)$ of equation (18) decreases with the evolution of the generations by equation (27) with f_{2i} = 0.80 and f_{2f} = 0.30;
- DEC1: uses $\alpha = 0.50$ and $\beta = 0.40$;
- DEC2: uses DEC2 with the constants given by f_{2i} = 0.80 and f_{2f} = 0.30;
- DEC3: uses DEC2 with the constants given by f_{2i} = 0.30 and f_{2f} = 0.80.

Table I summarizes results obtained by the DE and DEC implemented in this work, and those available in the literature, such as genetic algorithm (GA) and branch and bound (BB) [1] for the optimization of the supply chain.

TABLE I. RESULTS FOR THE OPTIMIZATION OF THE SUPPLY CHAIN USING DIFFERENT OPTIMIZATION METHODS.

	eval(*x*)			
Optimization methods	Best	Worst	Average	Standard Deviation
GA [1]	115495.00	-	-	-
BB [1]	113584.00	-	-	-
DE(1)	98368.90	106092.90	102861.70	3381.05
DE(2)	111139.60	118427.40	115931.18	3304.70
DE(3)	96544.50	99825.40	99825.40	1402.22
DEC(1)	**95025.00**	**96896.40**	**96322.60**	**601.99**
DEC(2)	127824.70	151550.30	141132.50	10054.09
DEC(3)	97421.40	107158.90	97421.40	4006.52

Table I shows that the best results were obtained using DE and DEC. GA did present acceptable results, but they still have to be improved. In table I is observed that the DE and DEC responded well for all the simulations attempts, except the

DEC(3) approach. The robustness of DE and DEC is higher that one of the tested GA and BB.

In table I is also observed that the DEC(1) obtains the best result, average and standard deviation of tested approaches. From 50 runs were made for each of the optimization methods involving 50 different initial trial solutions, it is shown that the results of DEC(1), DEC(2), DE(3) and DEC(4) approaches were significant in terms of best and average convergence. The best results in minimization of cost function, $eval(x)$, given by DEC(3) with reduction of 19.53% of the best result using GA of [1].

An important remark is that the EAs implemented in this work used the penalty-based method for constraint handling. In contrast, Mak and Wong [1] used a method that preserves feasibility of solutions by simply discarding unfeasible solutions generated during its GA evolution, at the expense of an extra computational overhead in the generation of populations. DE and DEC algorithms were implemented using Matlab 5.2, and took, in average, 330.99 and 332.18 seconds, respectively, to run in a PC-compatible with AMD Athon 1.0 GHz processor and 128 MB RAM.

V. Conclusion and Future Research

This paper presented a comparative study of DE and new DEC approaches for the optimization of a simplified supply chain. The supply chain was modeled as a mixed-integer programming problem, encompassing the optimization of costs related to stocking, manufacturing, transportation and shortage. The simplified supply chain had with 3 raw materials, 2 products, 3 retailers and 3 planning periods.

In this paper, the results obtained by DE and DEC are presented. According to data used in this work, is important to notice that shortage costs are very relevant. This is an attempt to reconcile two conflicting objectives: forecasted demand and low operational costs.

DE and DEC approaches obtained a better solution than those published in previous work of [1]. The results of these simulations are very encouraging and represent an important contribution to DE and DEC algorithm setups. DEC is employed in this paper to enhance the global exploration of traditional DE.

Future work will include the hybridization of the DE and DEC, by using a local search technique, such as branch-and-bound and simulated annealing. This approach, combining of the efficient global search of DE and DEC and the effectiveness of deterministic local search, possibly will give good results for real-world problems.

References

[1] K. L. Mak and Y. S. Wong, "Design of integrated production-inventory-distribution systems using genetic algorithm," in *Proccedings of Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, UK, pp. 454-460, 1995.

[2] H. -S. Hwang, "Design of supply-chain logistics systems considering service level," *Computers & Industrial Engineering*, vol. 43, 283-297, 2002.

[3] Y. H. Lee, C. S. Jeong, and C. Moon, "Advanced planning and scheduling with outsourcing in manufacturing supply chain," *Computers & Industrial Engineering*, vol. 43, 351-374, 2002.

[4] A. Syarif, Y. Yun, and M. Gen, " Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach," *Computers & Industrial Engineering*, vol. 43, 299-314, 2002.

[5] G. Zhou, H. Min, and M. Gen, "The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach," *Computers & Industrial Engineering*, vol. 43, 251-261, 2002.

[6] A. V. Smirnov, L. B. Sheremetov, N. Chilov, and J. R. Cortes, "Soft-computing technologies for configuration of cooperative supply chain, *Applied Soft Computing*, vol. 4, 87-107, 2004.

[7] R. Storn and K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report TR-95-012*, International Computer Science Institute, Berkeley, USA, 1995.

[8] R. Storn, "Differential evolution— a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[9] S. H. Strogatz, *Nonlinear dynamics and chaos*, Massachussetts, Perseus Publishing, 2000.

[10] L. S. Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on Power Systems*, 2006 (accepted for publication in issue of May 2006).

[11] H. -O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and fractals: new frontiers of science*, 2nd edition, Springer, New York, NY, 2004.