The Energy Minimization Method: A Multiobjective Fitness Evaluation Technique and Its Application to the Production Scheduling in a Petroleum Refinery

Mayron Rodrigues de Almeida

Sílvio Hamacher

Industrial Engineering Department
Catholic University of Rio de Janeiro
Rua Marquês de São Vicente, 225, Gávea,
Rio de Janeiro, RJ, Brasil, 22453-900
+55-21-529-9340
{mayron, hamacher}@rdc.puc-rio.br

Marco Aurélio Cavalcanti Pacheco

Marley B.R. Vellasco

ICA – Applied Computational Intelligence Laboratory
Electric Engineering Department
Catholic University of Rio de Janeiro
Rua Marquês de São Vicente, 225, Gávea,
Rio de Janeiro, RJ, Brasil, 22453-900
+55-21-529-9445
{marco, marley}@ele.puc-rio.br

Abstract – This paper reviews the multiobjective fitness evaluation method called Energy Minimization (Zebulum et al, 1998a; Zebulum et al, 1998b; Zebulum et al, 2000a), and presents an analysis of the method's behavior when used in a genetic algorithm applied to production scheduling of a petroleum refinery. The experimental results are presented and analyzed, leading to an overall evaluation of the benefits provided by the model.

1 Introduction

Frequently, real-world problems require simultaneous optimization of multiple performance objectives. Known optimization techniques, both conventional and non-conventional, such as genetic algorithms, are in general designed originally for single-objective problems, that is, problems in which the levels of optimality of the solutions can be given by the ordering of one single performance measure (e.g., a scalar value).

When multiple performance measures are necessary to rate a solution's optimality, it is necessary to define a way in which the different evaluations can be combined so as to provide a correct decision about which solution is better, and how much better it is. Without a suitable method to this decision, none of the otimization techniques can work efficiently in multiobjective optimization problems.

In this article, we first review in section 2, the main multiobjective optimization techniques, including the novel method called the energy minimization method, focus of this work. In Section 3, we discuss the energy minimization method's behavior and present the modified energy minimization method. Section 4 presents experimental results of the method when applied to the production scheduling in a petroleum refinery. Section 5 discusses these results and present final considerations.

2 Multiobjective Optimization Techniques

2.1 Dominance and the Pareto-Optimal Set

When comparing different solutions the concept of *dominance* plays an important role. This concept states that a given solution *v* dominates another solution *u* only if for no objective the evaluation of *v* is worse than that of *u*. Moreover, for at least one objective the solution *v* must present a better evaluation than that of *u* (Horn, 1997). For a hypothetical maximization problem of *n* objectives, with v_i and u_i corresponding respectively to the evaluations of solutions *v* and *u* for objective *i*, the dominance of *v* over *u* can be state as follows:

$$\begin{array}{l} \forall i \in \{1,...,n\}, v_i \geq u_i \cap \\ \exists i \in \{1,...,n\}, v_i > u_i \end{array} \qquad (1)$$

If a given solution is not dominated by any other, then that solution is said to belong to the *Pareto-optimal set*. This set actually corresponds to all those solutions that, in the absence of any other information about the problem, cannot be surely stated as being inferior to any other solution.

Several optimization methods for multiobjective problems seek the Pareto-optimal set, or *Pareto Frontier*, based on the fact that the optimal solution to the problem will surely belong to this set. The main problem with these methods is to find out which solution in the Pareto-optimal set is the most desirable one.

2.2 Linear Scalar Aggregation

The most simple and direct method for combining the multiple performance measures is the linear scalar-aggregative approach (Fonseca and Fleming, 1995; Horn, 1997). It basically consists of a *weighted sum* of the individual measures for each objective, with the final fitness evaluation *F* for a given solution being given by:

$$F = \sum_{i=1}^{n} w_i f_i \qquad (2)$$

where f_i corresponds to the fitness evaluation relative to objective *i*, and w_i corresponds to the respective weight, for a total of *n* objectives.

One important advantage of this method is being simple to implement, aside from being efficient computationally speaking. Its main disadvantage is the great difficulty in choosing appropriate weights w_i for a given problem.

In practice, the application of this method to a given problem almost always ends up in a costly and tiresome fine-tuning process, in which the algorithm is repeatedly tested with different sets of weights until satisfactory results are obtained.

2.3 Distance-to-Target Techniques

For many real-world problems a good solution must satisfy all the objectives at hand to a minimum extent. One way to achieve such a goal is to evaluate a given solution by calculating the distance between the vector composed of the individual measures f_i and the target-vector user made of ideal evaluations for each objective (Horn, 1997). Formally, such an evaluation method can be described by:

$$F = \left(\sum_{i=1}^{N} | user_i - f_i |^p \right)^{1/p} \quad p \geq 1 \qquad (3)$$

When *p=1,* we have the so-called *Manhattan or metropolitan distance*, which actually consists of a simple linear aggregation of the objectives combined with a target solution (Horn, 1997). With *p=2*, we obtain the more commonly used *Euclidean distance*. The quadratic form causes the solution to be more penalized for a value f_j far from the target value $user_j$ than it is benefited for having another value f_i close to its target $user_i$. There is now a "pressure to compromise" in order that it becomes harder for an unbalanced solution to be considered superior to a more compromising one.

Larger values for *p* will increase the penalty given to those solutions with mediocre performance for an objective. In the extreme case when $p \rightarrow \infty$, we obtain the technique known as *minimax* or *MinMax* (Horn, 1997), in which the evaluation of a given solution corresponds to the maximum distance of any of the *n* objectives relative to its target, that is:

$$F = \max\left(| user_i - f_i |\right) \qquad 1 \leq i \leq n \qquad (4)$$

This way, for a problem with two objectives the optimal solution will correspond to the exact intersection of the evaluation curves for each objective.

2.4 The Energy Minimization Method

The novel *Energy Minimization* (EM*)* (Zebulum et al, 1998a; Zebulum et al, 1998b; (Zebulum et al, 2000a; Zebulum et al, 2000b) tries to solve the main inconvenience of most scalar aggregation techniques, which is the choice of the weights associated with each objective. This method also incorporates the user's specifications, which is not trivially done with techniques that seek the Pareto-optimal set. The method, which has been designed for use within a genetic algorithm, is capable to adaptively update the weights throughout the evolutionary process. This way, greater priorities are constantly shifted to the objectives less satisfied by the population of solutions in general.

The EM method employs the linear scalar aggregation of the normalized fitness vector as follows:

$$F = \sum_{i=1}^{n} w_i Fnorm_i \qquad (5)$$

The normalization is usually implemented by the following equation:

$$Fnorm_i = \frac{f_i}{\bar{f}_i} \qquad (6)$$

where the denominator represents the fitness average for the population of solutions relative to objective *i*.

Weight updating is inspired on the back-propagation artificial neural network model (Churchland and Sejnowski 1992), and uses the formula below:

$$w_{i,t+1} = k_1.\alpha.w_{i,t} + k_2.(1-\alpha).e_{i,t} \quad (7)$$

In this equation, *t* specifies a particular generation of the evolutionary algorithm. Thus, $w_{i,t+1}$ is the weight value associated with objective *i* for the following generation and it is based on the current weight $w_{i,t}$ and an error measure $e_{i,t}$. k_1 and k_2 are normalization constants, computed by a procedure which will be described later. The idea of this weight-updating scheme is to assign larger weights to the objectives with larger errors. The error measure is calculated through the difference between the average performance of the population and the desired value specified by the user for objective *i* at a given time *t*.

$$e_{i,t} = \left| \frac{user_i - \bar{f}_{i,t}}{user_i} \right| \quad (8)$$

This way, the second term of equation (7) guarantees that the fitness function defined by equation (5) is dominated by those objectives with evaluations farthest from the desired values.

The first portion of equation (7) produces a effect analogous to the term *momentum* in the learning procedure of artificial neural networks, since it introduces memory to the system in a similar way. Its purpose is to increase the system's stability, avoiding drastic changes in the equation's outcome, which could make the genetic algorithm oscillate excessively. The constant α present in equation (7) is used to balance the two terms of this equation appropriately and can be assigned any value between 0 and 1.

The algorithm is initialized by choosing the starting values for the weights. The sum of these weights is defined by an integer value S_{w0} defined by the user:

$$S_{w0} = \sum_{i=1}^{n} w_{i,0} \quad (9)$$

The value of S_{w0} is arbitrary and does not influence the outcome of the system.

The purpose of normalization constants k_1 and k_2 is to allow the definition of a measure of the system's convergence state based on the sum of the weights $S_{w,t}$ for a given time *t*. Establishing an analogy with hopfield neural networks, the following scalar quantity is defined:

$$E = \sum_{i=1}^{n} w_i^2 \quad (10)$$

where *E* corresponds to the energy of the system. In fact, without considering the first term of equation (7), each weight $w_{i,t}$ is proportional to the corresponding error $e_{i,t}$. Considering that term, then it is necessary that the sum of the weights be proportional to the sum of the errors of the system at any given time *t*, that is:

$$S_{w,t} = k_3 \sum_{i=1}^{n} e_{i,t} = k_3 S_{e,t} \quad (11)$$

where:

$$k_3 = \frac{S_{w0}}{S_{e0}} \quad (12)$$

where S_{e0} corresponds to the sum of the errors observed for the first generation and k_3 is proportionality constant that takes into consideration the effect of the value chosen for S_{w0}. In order the sum of the weights to keep obeying the aforementioned relation, the following values must be assigned to the normalization constants:

$$k_1 = \frac{S_{w,t}}{S_{w,t-1}} \therefore k_2 = \frac{S_{w,t}}{S_{e,t}} \quad (13)$$

The computation of $S_{w,t}$ takes place before the calculation of the weight values themselves. Therefore, the energy obtained by equation (10) is guaranteed to yield a coherent measure of the state of the evolutionary process, the minimization of the system's energy actually corresponds to the satisfaction of multiple objectives. (Zebulum et al, 1998a; Zebulum et al, 1998b; Zebulum et al, 2000a; Zebulum et al, 2000b)

3 The Modified Energy Minimization Method

In the energy minimization method, the constant shifting of the priorities of the objectives often produces a *speciation* of the population, with groups of solutions specializing in the satisfaction of different sets of objectives. Thus, every time the weights are updated a different group of solutions (a different *species*) begins to dominate the remainder of the population. This way, the best solution yielded by the algorithm frequently oscillates between different solutions that excel in different sets of objectives, hardly ever converging to a final stable result.

Further studies need to be made in order to explain the output oscillations in the energy minimization method which could have been caused by the interaction between a non-convex Pareto surface and the inadequacies of weighted sum approach. Also, we have not studied the case where many points on the Pareto surface need to be identified.

In order to prevent the algorithm from oscillating and promote a convergence to a balanced compromise solution, with no particular objective being excessively neglected, Jonathan et al (2000) have proposed a modified form of the energy minimization method.

The first change is on fitness evaluation function described in equation (5). The new fitness measure considers the distance betweeen the solution's vector of evaluations and the target vector of user-defined values for each objective (see Section 2.3).

Thus, formally we rewrite equation (5) and combine it with equation (3):

$$F = \left(\sum_{i=1}^{N} w_i . e_i^p \right)^{1/p} \quad p \geq 1 \qquad (14)$$

where e_i corresponds to the error of the solution's evaluation relative to its target value for objective i, and p defines how the vector distance is measured, as seen in section 2.3.

By employing $p=2$, we define a solution's final fitness as the *quadratic weighted sum of the errors* of its individual evaluations for each objective. By increasing the value of p we also increase the "pressure to compromise" induced by the algorithm (see Section 2.3). Consequently, the speciation tendency observed for the original method is averted, preventing the evolutionary process from excessively oscillating between different species

This fitness evaluation is only coherent when the errors e_i are normalized so that they can be appropriately compared to each other. Indeed, it must also be observed that a correct normalization should consider each of the evaluations f_i for a given objective i relative to its *search space*. Thus, for instance, the best possible evaluation for a particular objective i could yield $e_i = 0$, with the worst possible value corresponding to $e_i = 1$.

When the search space is not known *a priori*, it is necessary to *estimate* it during the evolutionary process itself, using the best and worst evaluations found so far ($best_i$ and $worst_i$). The average evaluation avg_i of the population is employed to extract the notion of a "reasonable" evaluation for the current stage of the evolution. Optionally, the target values $user_i$ is used in order to achieve a more controlled estimate of the search space, although these values can somewhat distort the evolutionary process due to their arbitrary nature.

Considering these aspects, the error normalization equation has been changed to:

$$e_i = \left| \frac{best_i - f_i}{best_i - avg_i} \right| \qquad (15)$$

where $best_i$ corresponds to the best evaluation found so far for objective i or a target defined by the user, while avg_i corresponds to the average evaluation obtained for the entire population relative to this objective.

4 Applying the Modified Minimization Method to Production Scheduling of a Petroleum Refinery

This Modified Energy Minimization Method was successful implemented and tested on an evolution of electrical circuit problem by Jonathan et al (2000). In order to prove the performance of the multiobjective optimization technique defined previously, the algorithm has been tested using genetics algorithms to solve the production scheduling in a real plant of the fuel oil and asphalt area of a petroleum refinery.

4.1 Description of the Fuel Oil and Asphalt Area at REVAP

The Henrique Lage Refinary (REVAP) is located in São José dos Campos, in the State of São Paulo. The refinary receives crude and intermediate feedstock from pipelines and dispatches most of its finished products via pipelines as well.

The key business goals to maximize diesel and jet fuel production at minimum cost subject to the constrained of supplying the market demand for all the remaining products, mainly fuel oil and asphalt.

The refinary total capacity is 1.000.000 m^3/m. The production of the fuel oil and asphalt area are 180.000 m^3/mês and 40.000 m^3/mês, respectively.

Magalhães et al (1998) state that the fuel oil and asphalt area is very complicated to schedule because of the several options for components and final products existent in this area.

The problem of production scheduling relative for the fuel oil and asphalt production at REVAP is a real problem based on a multiproduct plant, with one machine (mixer), without setup time and with resource-constrained in continuous operating. The plant has one desasphalting unit (UDASF), one Fluid Catalytic Cracking Unit (UFCC), six tanks to stock diluents (22, 23, 24, 25, 26 and 27), twenty one tanks to stock final products (11, 12, 13, 14, 15, 16 - fuel oil; 1, 2, 3 , 4, 5 – asphalt; 6, 7, 8, 9 – ultraviscous fuel oil; and 10, 18, 19 , 20, 21 – CM-30), five terminal of truck-tanks, two pipelines to transport final produtcts to final consumer and connections between this system's elements, showed at Figure 1.

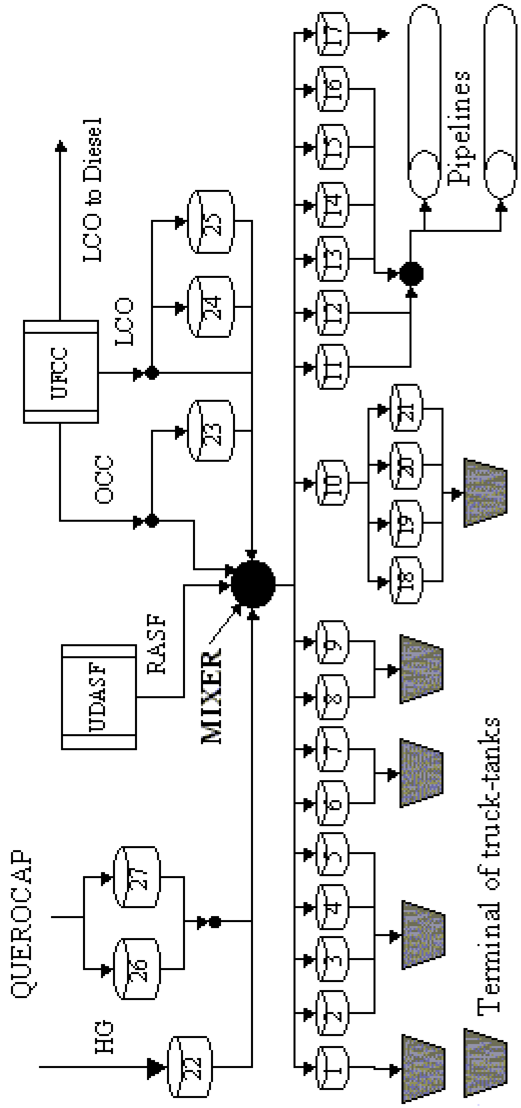

Figure 1 - Schema of the fuel oil and asphast area

During the time interval of production scheduling, asphaltic residue (RASF) is produced continuously by desasphalting unit (UDASF). This RASF is diluted with decanted oil (OCC) and/or light cocking oil (LCO) to produce four types of fuel oil (OC-1A/B, OC-2A/B, OC-3A/B and OC-Bunker) and two types of ultraviscous fuel oil (OC-7A/B and OC-9A/B), diluted with high gasoil (HG) and QUEROCAP to produce CM-30 or then dilute with HG to produce asphalt (CAP-07 and CAP20).

Operational constrains that must be considered in this problem, are:

- No tank can ever be loaded and unloaded simultaneously;

- Production must be continuous, because RASF never stop to flow to the mixer;
- Demand must be provided at a pre-fixed schedule, i.e., it isn't the scheduler's task to fix the best demand's schedule;
- There is a minimum volume that must rest at final product and intermediate product tanks;
- There is a preparation period for the final products (period of 12 hours to mix final products) and a quality control period (16 hours to asphalt, 16 hours to CM-30 and 8 hours to the others).

4.2 Experimental Results

To analyze the experimental results obtained by GA, we used data from Table 1 as problem starting values. Table 2 presents the GA parameters. We tested 4 scenarios with different starting values for the final product tanks (Table 1) and one real demand scenario that represents 165,32 hours or 98,4 percent of production with maximum operation capacity.

Throughout the genetic algoritm, each scheduling is simulated and has its performance estimated by some key value drivers, namely: *non-provided demand, production that can't be allocated in the tanks, number of operational mode changes* and *average inventory*. The tests were done in two stages. In the **first stage**, the targets for the tests were four: minimize *non-provided demand*, minimize *production that can't be allocated in the tanks*, minimize *number of operational mode changes* and minimize *average inventory.* And, in the **second stage**, only the first three targets were attended.

Tanks	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Percents from operation maximum volume of the sum of every tanks	51%	35%	24%	19%

Table 1 – Starting values scenarios of final products tanks.

The motivation for the different scenarios of starting values in final product tanks is to find, experimentally, the best inventory level, attending the other key values drivers targets safely.

Solution method	order
Chromosome Representation	list of tasks
Population	80
Generation	125
Crossover	80% - 50%
Mutation	10% - 50%
Linear normalization	yes
Steady State without duplicates	yes
GAP	0,8
Interval of weight's actualization	3 generations
Numbers of experiments	6

Table 2 – GA parameters

Table 3 presents the average result obtained in the last generations of each scenario in 6 experiments. As expected, tests results from the second stage are, in general, better than the results from the first stage. In fact, the minimization of average inventory gives always better results in the tests from the first stage. Nevertheless, all the other key values drivers worsen their performance in this situation.

SCENARIOS	OBJECTIVES	Number of operational mode changes	Average inventory (m^3)	Non-provided demand (m^3)	Production that can't be allocated in the tanks (m^3)
1	**4**	26,17	140700	585,67	158,67
	3	23,67	142190	338	39,67
2	**4**	24	104570	512,67	0
	3	23	105640	0	0
3	**4**	27,67	78466	192	79,33
	3	25,17	80310	89,67	39,67
4	**4**	24,33	67563	833,33	146
	3	28	68920	612,67	0

Table 3 – Average results obtained in the last generation of each scenario in 6 experiments.

The next figures show GA evolution, during 125 generations of individuals, at every scenarios, in 6 experiments. The Figure 2, Figure 3, Figure 4 and Figure 5 present the GA evolution graphics for the tests from the first stage. The Figure 6, Figure 7, Figure 8 and Figure 9 present the GA evolution graphics for the tests from the second stage. The Figure 10 presents in detail (its same curves shows up at Figure 5 and Figure 9) the evolution's behaviour of minimization of average inventory in Scenario 2 for the first and second stage of tests.

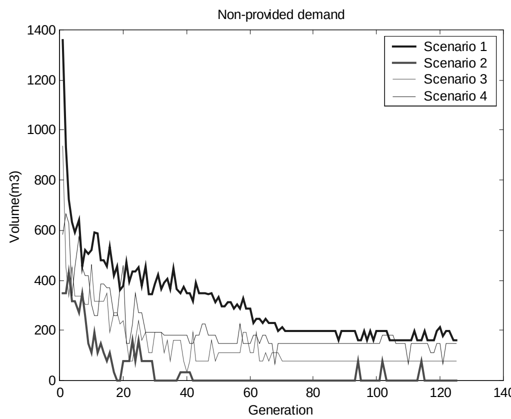

Figure 2 - Non-provided demand / 4 Objectives

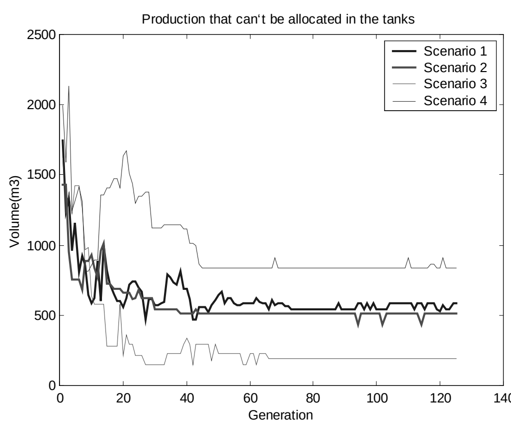

Figure 3 - Production that can’t be allocated in the tanks / 4 Objectives

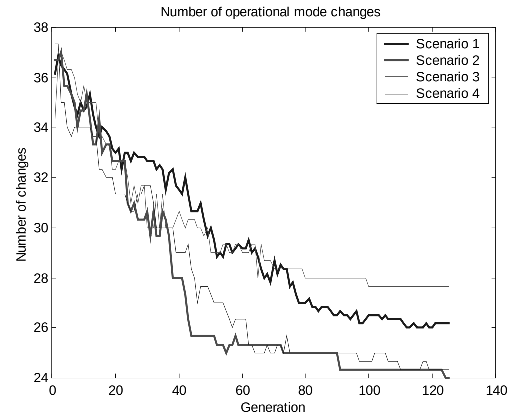

Figure 4 - **N**umber of operational mode changes / 4 Objectives

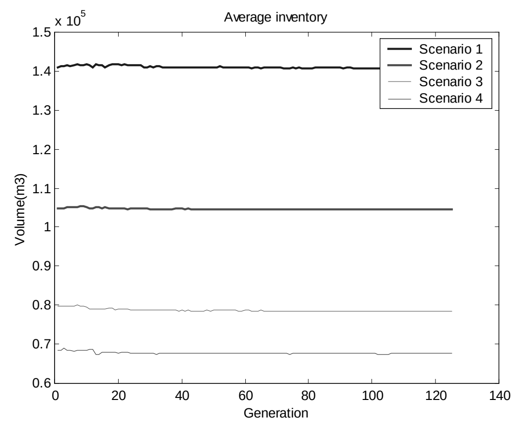

Figure 5 - Average inventory / 4 objectives

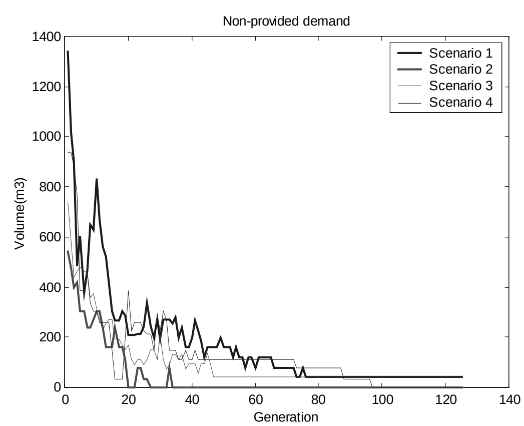

Figure 6 - Non-provided demand / 3 Objectives

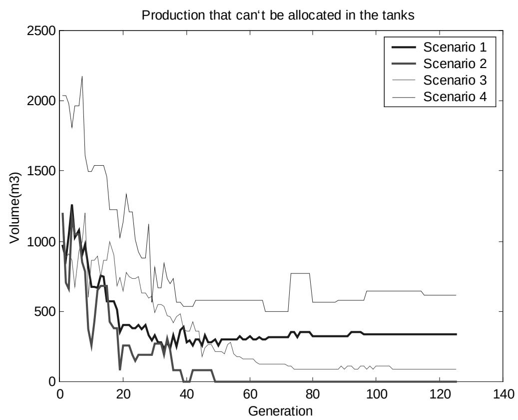

Figure 7 - Production that can’t be allocated in the tanks / 3 objectives

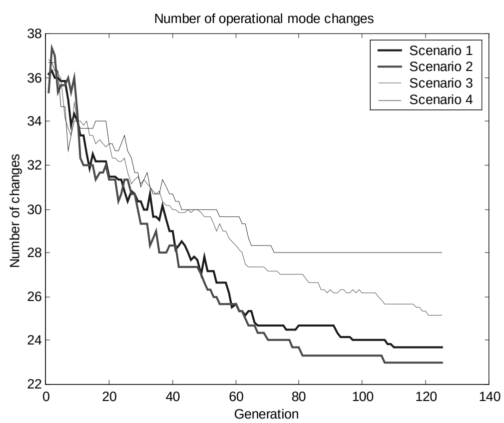

Figure 8 - **N**umber of operational mode changes / 3 Objectives

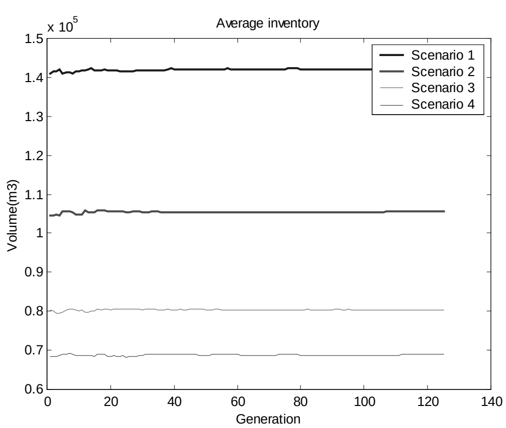

Figure 9 - Average inventory / 3 objectives

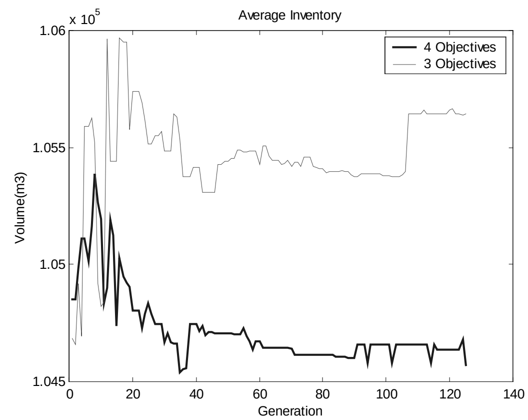

Figure 10 – Average inventory from scenario 2 with 3 and 4 objectives

5 Conclusions

In this paper we have presented a multiobjective optimization technique, used with genetics algoritms, to solve a production scheduling integrating a lot-sizing and sequencing of production in a real plant of the fuel oil and asphalt area of a petroleum refinery.

After analyzing the experimental results, it can be conclued that approach used had a positive effect on the algorithm as a whole. The GA had excellent performance (many times near from the optimum[1]) to *non-provided demand objective* and *production that can’t be allocated in the tanks objective*; provided satisfactory results (in agreement with real scheduling of refinery) to *numbers of operational mode changes objective* and had insignificantly results in practical terms to *average inventory objective* for each scenario. However, the performance is satisfactory between scenarios, i.e., it’s possible to find a operation level for inventories, that for same time, has zero penalty to *non-provided demand* and *production that can’t be allocated in the tanks* and minimize the *numbers of operational mode changes objectives.*

We can conclude throughout the tests realized by Jonathan et al (2000) and the tests realized in this paper that the modified energy minimization method increase the genetic algoritm capacity to find better solutions at problems envolving multiobjectives.

6 References

Churchland, P.S., Sejnowski, T. J., “The Computacional Brain”, MIT Press, 1992.

Fonseca, C.M., Fleming, P.J., “An Overview of

[1] Optimum referring to find a scheduling that has evaluation with zero penalty to *non-provided demand objective* and *production that can’t be allocated in the tanks objective.*

Evolutionary Algorithms in Multiobjective Optimization”, Evolutionary Computation, Volume 3, Number 1, pp. 1-16, MIT Press, Spring, 1995.
Horn, J., “Multicriterion Decision Making”, in Handbook of Evolutionary Computation, IOP Publishing LTD and Oxford University Press, F1.9, 1997.
Jonathan, M., Zebulum, R. S., Pacheco, M. A., Vellasco, M. “Multiobjective Optimization Techniques: A Study Of Energy Minimization Method And Its Application To The Sysnthesis Of Ota Amplifiers”, “The Second NASA/DoD Workshop on Evolvable hardware”, pp. 133-140, IEEE Computer Society, Los Alamitos, Silicon Valley, California, USA, 2000.
Magalhães, M.V.O, Moro, L.F.L, Smania, P., Hassimoto, M.K., Pinto, J.M, Abadia, G.J., SIPP – A solution for Refinery Scheduling. In: “NPRA Computer Conference”, San Antonio (EUA), 1998.
Zebulum, R.S., Pacheco, M. A., Vellasco, M., “Variable Length Representation in Evolutionary Electronics”, Evolutionary Computation, Volume 8, Number 1, pp.93-120, MIT Press, Spring, 2000.
Zebulum, R.S., Pacheco, M.A., Vellasco, M., “A Novel Multi-Objective Optimisation Methodology Applied to Synthesis of CMOS Operational Amplifiers", Journal of Solid-State Devices and Circuits, Microelectronics Society - SBMICRO, 2000.
Zebulum, R.S., Pacheco, M.A., Vellasco, M., “Analog Circuits Evolution in Extrinsic and Intrinsic Modes”, in the Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98), Lausanne, Switzerland, September, 23-26, 1998. M.Sipper, D.Mange and A. Pérez-Uribe (editors), vol. 1478, pp. 154-165, LNCS, Springer-Verlag, 1998.
Zebulum, R.S., Pacheco, M.A., Vellasco, M., ”A Multi-Objective Optimisation Methodology Applied to the Synthesis of Low-Power Operational Amplifiers”, proceedings of the XIII International Conference in Microelectronics and Packaging, Vol. 1, Ivan Jorge Chueiri and Carlos Alberto dos Reis Filho (Eds), pp. 264-271, Curitiba, Brasil, August, 1998.