Decision-making Tools and Memetic Algorithms in Management and Linear Programming Problems

Jesús M. Larrañaga LESACA[1]
Ekaitz Zulueta GUERRERO
Fernando Elizagarate UBIS
Jon Alzola BERNARDO

Abstract

Operational Research uses a set of tools based on scientific research principles to achieve rational and meaningful management decisions. This article tries to give solution to a highly complex Linear Programming problem by using Simplex method, Solver and a hybrid prototype which combines the theories of Genetic Algorithms with a new local search heuristic technique. Hybridization of these two techniques is becoming known as Memetic Algorithm. Additionally, this article tries to present different techniques to support management decision-making, with the intention of being used increasingly in the business environment sustaining, thus, decisions by mathematics or artificial intelligence and not only by experience.

Keywords: *quantitative management, quantitative methods, decision-making, linear programming, operational research, heuristics, hybrid methods, memetic algorithms*

JEL classification: D81, E27

1. Introduction

Operational Research is a modern scientific discipline which uses theory, methods and special techniques to look for the solution of management and decision making problems. To find the solution, Operational Research generally represents the problem as a mathematical model, which is analyzed and evaluated previously. It is necessary to have enough information to develop a model, faithful to reality. Otherwise decisions would be made through experience or the model would be established through the simulation of production processes (Villanueva, 2008).

The most important objective in Operational Research is to support the "optimal decision making."

[1] **Jesús M. Larrañaga LESACA,** Dpto. de Organización de Empresas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. University of the Basque Country, Vitoria-Gasteiz, Spain
Ekaitz Zulueta GUERRERO, Dpto. de Ingeniería de sistemas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. University of the Basque Country, Vitoria-Gasteiz, Spain
Fernando Elizagarate UBIS, Dpto. de Organización de Empresas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. University of the Basque Country, Vitoria-Gasteiz, Spain
Jon Alzola BERNARDO, Tecnalia Research & Innovation, Miñano, Spain

In this paper, “Juice Processing Problem” is studied, as a high complex problem. The problem statement can be found in two books: Investigación Operativa. Programación lineal y aplicaciones de Sixto Ríos (1996) and Problemas de Investigación Operativa, Sixto Ríos (Ra-Ma 2006); but it is not solved. So, the aim of this paper is to give solution to this problem using classical optimization methods as Simplex method through www.PHPSimplex.com free tool and using Solver tool from Microsoft Excel. Finally, it will be described a prototype developed in C++ language using an hybrid method, known as Memetic Algorithm, that combines Genetic Algorithm theories and an heuristic created to improve the prototype, reaching optimal solutions.

Before solving the “Juice Processing Problem”, three techniques will be proved with an easier problem: “The Producer of Beer Problem” (Sixto Ríos, 1996).

2. Optimization techniques

There are several alternatives to solve a complex problem that maximizes or minimizes a linear function subject to linear constraints. In this article we will use three.

2.1 Simplex method

Simplex method, developed by the mathematician George Bernard Dantzig in 1947, is a popular technique to give numerical solutions to linear programming problems that involve three or more variables.

Matrix algebra and the process of Gauss-Jordan elimination to solve a system of linear equations are the basis of the simplex method. Solving linear programs by the simplex method involves making lots of calculations through successive tables, especially when the number of variables and / or restrictions is relatively high. In real cases, the magnitude of the problems becomes necessary to use computers. The web www.PHPSimplex.com allows to solve problems online directly or step by step seeing how Simplex tables change.

2.2 Solver tool of Microsoft Excel

Solver is a tool to solve and optimize equations using numerical methods. Solver can be used to optimize functions of one or more variables, with or without restrictions. EXCEL Solver option is used to solve linear and nonlinear optimization problems. With Solver it can be solved problems with up to 200 decision variables, 100 explicit and 400 simple restrictions (upper and lower bounds and integer restrictions on decision variables.) To access Solver, select "Tools" from the main menu and then “Solver”.

How to use the Solver tool: The “Solver Parameters” window is used to describe the optimization problem to Excel. “Set Target Cell” field contains the cell where the objective function for the problem is. If you want to find the maximum or minimum, select Max or Min. The dialog box “By Changing Cells” will contain the location of the decision variables for the problem. Finally, the restrictions must be specified in the “Subject to the Constraints” field by clicking Add. “Change” button makes it possible to modify the introduced restrictions and “Delete” serves to erase the previous restrictions. “Reset All” clears the current problem and restore all settings to their default values.

2.3 Genetic Algorithms, Heuristics and Memetic Alogrithms

In the 70's, it was developed a new search technique, known as Genetic Algorithms (Holland, 1975) which was based on the theory of evolution (Darwin, 1859). Genetic Algorithms select the best possible solutions until reaching the optimal solution, using different methods based on nature, such as selection, crossover or mutation in order to improve the solutions or individuals to the global optima. The basic principles of genetic algorithms are well described in numerous texts (Davis, 1991), (Michalewicz, 1996), (Whitley, 1994).

Heuristics are techniques, based on experience, to solve a problem. These rules are used when it is needed to reach a good solution in a reasonable time or when there is not a method capable to reach optimal solutions, satisfying the constraints of the problem. Information about heuristics can be found in various articles (Michalewicz and Fogel, 2004), (Pearl, 1984) (Chica, et al., 2009).

Memetic Algorithms arise as a combination of Genetic Algorithms and heuristics, normally based on local search. Individuals created by both techniques compete and cooperate completing a synergy (Moscato, 1989) and they obtain very good results (Cotta, 2007).

3. The Producer of Beer Problem

3.1 Statement

A brewery produces three types of beer called stout, lager and low alcohol. It is necessary to obtain them: water and hops, both with no limit, and malt and yeast, which limits the daily production capacity. The following table shows the required amount of each of these resources to produce a litre of each beer, available kilogram’s of each resource and benefits in monetary units (mu) per litre of each beer produced. The producer's problem is to decide how much to produce of each beer in order to maximize the daily total benefit.

Table 1 Summary of the problem statement

	Stout	**Lager**	**Low alcohol**	**Availability**
Malt	2	1	2	30
Yeats	1	2	2	45
Benefit	4	7	3	

3.2 Problem model

Decission variables:

- X1 = Production of stout beer (litres per day).
- X2 = Production of lager beer (litres per day).
- X3 = Production of low alcohol beer (litres per day).

Constraints:

- $2X1 + X2 + 2X3 \leq 30$
- $X1 + 2X2 + 2X3 \leq 45$

Benefit maximization fuction:

$Max \;\; z = 4X1 + 7X2 + 3X3$

4. Giving solution to "The Producer of Beer Problem"

4.1 Simplex method

The web www.PHPSimplex.com will help us to find solutions with the Simplex method. It contains an online free tool for solving linear programming problems.

The model of the problem must be entered in the tool to reach the optimal solution and compare it later with the solutions obtained using Solver and the prototype based on Memetic Algorithms.

First, we choose the Simplex method and enter the number of variables and constraints, three and two respectively, for this problem, and click on the button "Continue." See Figure 1.

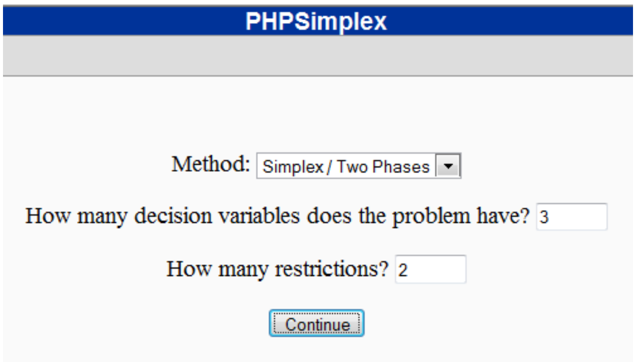


Figure 1 First step to solve the problem using PHPSimplex

Then, we choose "Maximize" because we are looking for the maximum benefit for this problem and we fill the gaps with the respective coefficients defined in the problem model. After it, we can click "Continue". See Figure 2.

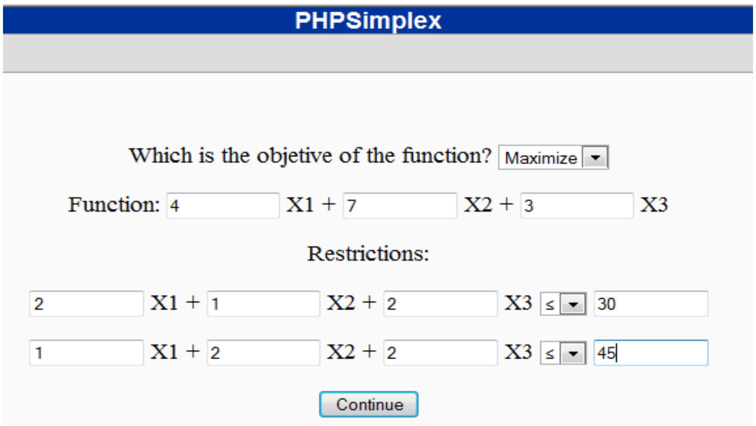


Figure 2 Second step to solve the problem using PHPSimplex

After introducing the problem model, the tool shows it in the standard form, adding slack, surplus and artificial variables as it can be seen in Figure 3.

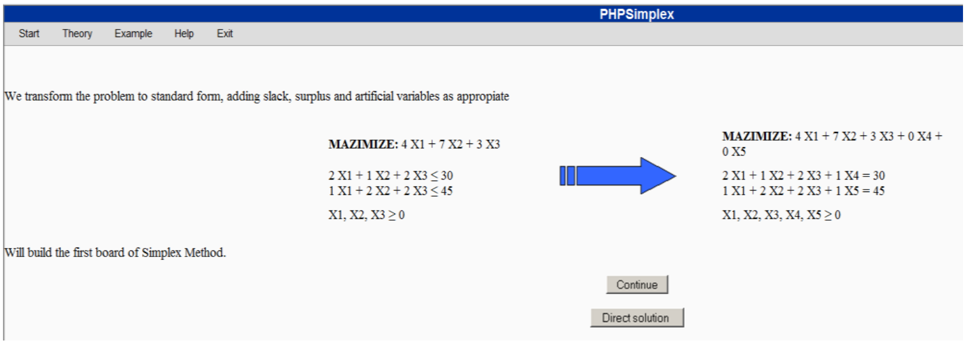


Figure 3 Third step to solve the problem using PHPSimplex

Finally, we click on "Direct Solution" button to find the optimal decision that maximizes the benefit. See Figure 4.

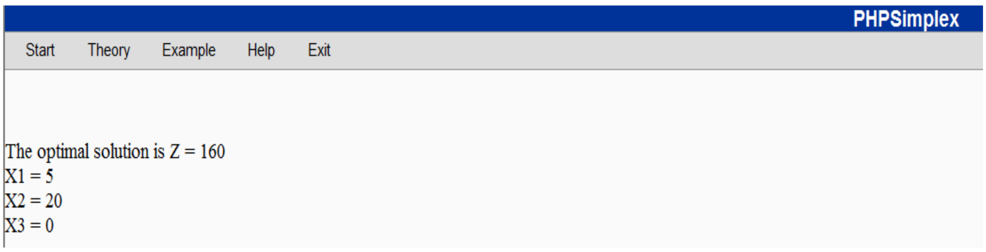


Figure 4 Fourth step to solve the problem using PHPSimplex.

4.2 Solver tool of Microsoft Excel

First, the data model of the problem is entered in Excel in order to apply the Solver tool, based on quasi-Newton method or conjugated gradient algorithm. Quasi-Newton method normally needs more memory but less number of iterations than the conjugate gradient method. The result, as seen in Figure 5 is the same as obtained using the Simplex method.

Objetivo z	**160**				
Variables decisión	x1	x2	x3		
	5	20	0		
Coeficientes cj	**c1**	**c2**	**c3**		
	4	7	3		
Restricciones				**Formula**	**b**
1	2	1	2	30	30
2	1	2	2	45	45

Figure 5 Optimal solution found by using Solver tool of Microsoft Excel 2007

4.3 Memetic Algorithm or Hybrid Method that combines Genetic Algorithm and Heuristics

It has been built a prototype based on Genetic Algorithms performing a lot of tests in order to define different kinds of mutation operators, crossovers, population sizes, replacements… These tests have served to select the methods, operators and values that obtain the best results. The prototype performance is excellent, except for solutions that require integer values because the prototype works with real numbers. Therefore, it has been implemented a local search heuristic called that adjust the best solution by changing the real values of the variables for the nearest integers, creating new individuals who also compete to enter into the population. The results have been excellent, as it is shown in Figure 6.

The algorithm has the following features:

- Random generation of the initial population.
- Number of evaluated individuals = 1000.
- Mutation probabilities = 20 %.
- Population size = 20.
- Uniform crossover (Syswerda, 1989).
- The Fitness Value is the maximum value of z (benefit).
- Roulette selection (Michalewicz, 1996).
- Worst Among Most Similar Replacement (WAMS) (Shuhei, 2003). Replacement based on the euclidean distance between two individuals in order to maintain diversity in the population. If two individuals are similar (Euclidean

distance shorter than 0.01 units), only the one with higher fitness value will exist in the population. Replace Worst Strategy (RW) for Individuals that satisfy the minimum distance. The worst individual in the population will be replaced, maintaining the population size.

- Local search heuristic that modifies the best individual of the population in each generation changing the real values to integer values.

The prototype follows these steps:

1. Initialization or initial population generation.
2. Fitness function computation for each individual.
Repeat
3. Application of selection operator (Roulette) to obtain two parents.
4. Application of crossover and mutation operators.
5. Application of the heuristic that searches Integer Values.
6. Fitness function computation for the obtained offspring.
7. WAMS and RW replacement.
Until stop criterion is reached.

The following figures show the evolution of the solutions to the global optima.

In Figure 6, it is shown the randomly generated initial population to solve the problem. The first three columns correspond to X1, X2 and X3, respectively, and the fourth column is the fitness value.

Initial Population				
1st Individual:	8.04	9.64	2.13	106.03
2nd Individual:	3.33	8.76	6.93	95.43
3rd Individual:	7.85	8.23	1.39	93.18
4th Individual:	5.39	9.69	0.44	90.71
5th Individual:	1.96	9.66	4.95	90.31
6th Individual:	7.73	7.72	1.67	89.97
7th Individual:	9.21	7.32	0.37	89.19
8th Individual:	2.54	8.25	7.02	88.97
9th Individual:	0.99	8.68	7.03	85.81
10th Individual:	1.69	7.95	7.49	84.88
11th Individual:	4.96	6.7	5.72	83.9
12th Individual:	6.75	7.31	1.74	83.39
13th Individual:	5.9	6.62	4.3	82.84
14th Individual:	2.88	6.89	7.54	82.37
15th Individual:	1.95	9.21	2.9	80.97
16th Individual:	6.19	7.93	0.21	80.9
17th Individual:	3.5	7.21	5.46	80.85
18th Individual:	0.45	9.13	5.01	80.74
19th Individual:	1.06	8.75	4.5	78.99
20th Individual:	0.03	8.6	5.85	77.87

Figure 6 Randomly generated initial population

It can be seen in Figure 7, the evolution of the best solution to reach the optimal solution. It also shows how the heuristic method has worked in last generation, reaching quickly the optimal solution.

6	11	3	110
6.1251	13.1777	2.1	123.044
6.2511	15.2777	1.1	135.248
6.3611	15.2777	0.99	135.358
5.4705	16.3927	1.09	139.901
5.6621	16.4927	1.09	141.367
4.7621	17.6181	1.2669	146.176
4.8621	17.7181	1.2669	147.276
4.8621	17.7181	1.2669	147.276
4.9621	19.7181	0.1669	158.376
4.9649	19.7181	0.1669	158.387
4.9649	19.7181	0.1669	158.387
4.9649	19.7181	0.1669	158.387
4.9649	19.7181	0.1669	158.387
4.9649	19.7181	0.1669	158.387
4.9649	19.7181	0.1728	158.405
4.9649	19.7181	0.1728	158.405
4.9254	19.8393	0.1396	158.995
4.9254	19.8393	0.1396	158.995
5	20	0	160

Figure 7 Evolution of the best individual to reach the optimal solution

Figure 8 shows the final population. The optimal solution recommended will correspond to the first Individual: X1 = 5, X2 = 20, X3 = 0, with a benefit of 160 m.u.

1st Individual:	5	20	0	160
2nd Individual:	4.9254	19.8393	0.1396	158.995
3rd Individual:	4.8313	19.9178	0.0428999	158.879
4th Individual:	4.8269	19.7181	0.1728	157.853
5th Individual:	4.9254	19.6673	0.1396	157.791
6th Individual:	4.71359	19.7181	0.15021	157.332
7th Individual:	4	20	0	156
8th Individual:	4.9254	19.368	0.1396	155.696
9th Individual:	4.9649	19.1311	0.1669	154.278
10th Individual:	4.951	19.1661	0.0470999	154.108
11th Individual:	5.0444	18.9783	0.2666	153.826
12th Individual:	5	19	0	153
13th Individual:	4.9649	18.9091	0.1669	152.724
14th Individual:	4	19	1	152
15th Individual:	4.28589	19.0943	0.2669	151.604
16th Individual:	4.8621	18.7181	0.1669	150.976
17th Individual:	4.8621	18.5742	0.3669	150.569
18th Individual:	4.7621	18.6181	0.2669	150.176
19th Individual:	4	19	0	149
20th Individual:	4.8621	17.7181	1.2669	147.276

Figure 8 Final population

The tree optimization techniques work properly with this problem. Now they will be proved with a high complex problem, where some of them will find some difficulties to reach the optimal solution.

5. “The Juice Processing Problem”

5.1 Statement

A food company produces pear, orange, lemon, tomato and apple juices. They also produce two other types called H and G which combine some of the mentioned before. The availability of fruit for the next period, the production costs and the selling prices for the simple fruit juices, are given in Table 2. Table 3 shows the specifications of combined fruit juices.

Table 2 Specifications of simple fruit juices

Fruit	Maximum availability (Kg)	Coste (cents/Kg)	Selling price (cents/Kg)
Orange (N)	32.000	94	129
Pear (P)	25.000	87	125
Lemon (L)	21.000	73	110
Tomato (T)	18.000	47	88
Apple (M)	27.000	68	97

Table 3 Specifications of combined fruit juices

Combined fruit juice	Specification	Selling price (cents/Kg)
H	No more than 50% of M No more than 20% of P Not less than 10% of L	100
G	40% of N 35% of L 25% of P	120

The demand of the different fruit juices is high, so it is intended to sell the whole production. One kg of fruit will be one litre of juice. The aim is to formulate a linear program to determine the production levels of the seven juices in order to have maximum benefit.

5.2 Problem model

The number of constraints has been reduced from 13 to 11, so H and G disappear as variables being substituted by the quantity of each simple juice that form the combined one: Hm, Hp, Hl, Gn, Gl, Gp (H = Hm + Hp + Hl y G = Gn + + Gl + Gp).

So the variables are N (Orange), P (Pear), L (Lemon), T (Tomato), M (Apple), Gn (Quantity of orange in G), Gl (Quantity of lemon in G), Gp (Quantity of pear in G), Hm (Quantity of apple in H), Hp (Quantity of pear in H), Hl (Quantity of lemon in H).

Constraints:

$$
\begin{aligned}
&N + Gn \leq 32000 \\
&P + Hp + Gp \leq 25000 \\
&L + Hl + Gl \leq 21000 \\
&T \leq 18000 \\
&M + Hm \leq 27000 \\
&Gn = 0'4(Gn + Gl + Gp) \\
&Gl = 0'35(Gn + Gl + Gp) \\
&Gp = 0'25(Gn + Gl + Gp) \\
&Hm \leq 0'5(Hm + Hp + Hl) \\
&Hp \leq 0'2(Hm + Hp + Hl) \\
&Hl \geq 0'1(Hm + Hp + Hl)
\end{aligned}
$$

Benefit maximization function:

$$Max$$
$$z = 35N + 38P + 37L + 41T + 29M + 120(Gn + Gl + Gp) - 94Gn - 73Gl - 87Gp + \\ + 100(Hm + Hp + Hl) - 68Hm - 87Hp - 73Hl$$

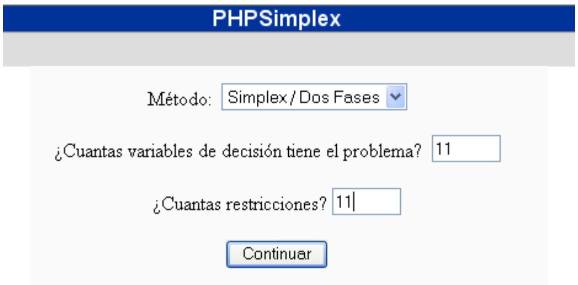
Simplifying:

$$Max$$
$$z = 35N + 38P + 37L + 41T + 29M + 26Gn + 47Gl + 33Gp + 32Hm + 13Hp + 27Hl$$

6 Giving solution to "The Juice Processing Problem"

6.1 Simplex method

The model has been entered in PHPSimplex as it is shown in Figure 9.



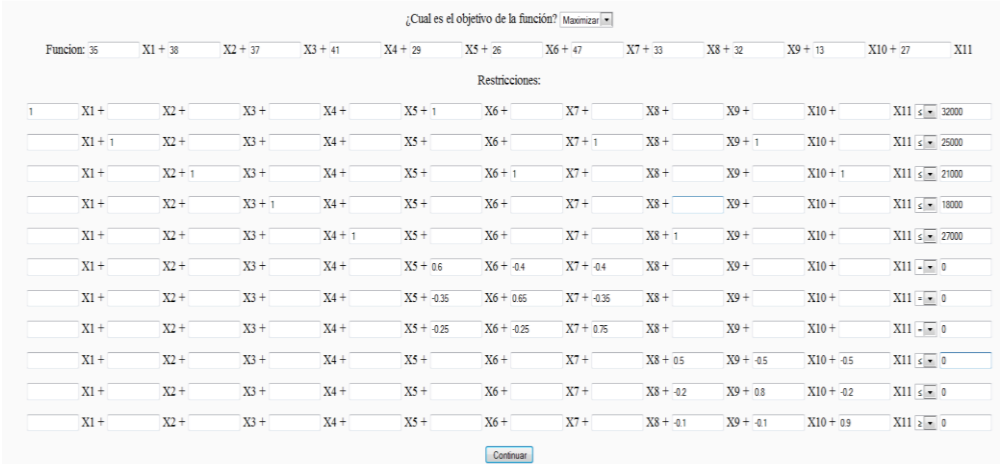


Figure 9 How to solve the problem using www.PHPSimplex.com

Figure 10 shows the model in the standard form, adding slack, surplus and artificial variables.

MAXIMIZAR: 35 X1 + 38 X2 + 37 X3 + 41 X4 + 29 X5 + 26 X6 + 47 X7 + 33 X8 + 32 X9 + 13 X10 + 1 X11

1 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 + 1 X6 + 0 X7 + 0 X8 + 0 X9 + 0 X10 + 0 X11 ≤ 32000
0 X1 + 1 X2 + 0 X3 + 0 X4 + 0 X5 + 0 X6 + 0 X7 + 1 X8 + 0 X9 + 1 X10 + 0 X11 ≤ 25000
0 X1 + 0 X2 + 1 X3 + 0 X4 + 0 X5 + 0 X6 + 1 X7 + 0 X8 + 0 X9 + 0 X10 + 1 X11 ≤ 21000
0 X1 + 0 X2 + 0 X3 + 1 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 + 0 X9 + 0 X10 + 0 X11 ≤ 18000
0 X1 + 0 X2 + 0 X3 + 0 X4 + 1 X5 + 0 X6 + 0 X7 + 0 X8 + 1 X9 + 0 X10 + 0 X11 ≤ 27000
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 + 0.6 X6 -0.4 X7 -0.4 X8 + 0 X9 + 0 X10 + 0 X11 = 0
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 -0.35 X6 + 0.65 X7 -0.35 X8 + 0 X9 + 0 X10 + 0 X11 = 0
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 -0.25 X6 -0.25 X7 + 0.75 X8 + 0 X9 + 0 X10 + 0 X11 = 0
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 + 0.5 X9 -0.5 X10 -0.5 X11 ≤ 0
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 -0.2 X9 + 0.8 X10 -0.2 X11 ≤ 0
0 X1 + 0 X2 + 0 X3 + 0 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 -0.1 X9 -0.1 X10 + 0.9 X11 ≥ 0

X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11 ≥ 0

MAXIMIZAR: 35 X1 + 38 X2 + 37 X3 + 41 X4 + 29 X5 + 26 X6 + 47 X7 + 33 X8 + 32 X9 + 13 X10 + 1 X11 + 0 X12 + 0 X13 + 0 X14 + 0 X15 + 0 X16 + 0 X17 + 0 X18 + 0 X19 + 0 X20 + 0 X21 + 0 X22 + 0 X23

1 X1 + 1 X6 + 1 X12 = 32000
0 X1 + 1 X2 + 1 X8 + 1 X10 + 1 X13 = 25000
0 X1 + 1 X3 + 1 X7 + 1 X11 + 1 X14 = 21000
0 X1 + 1 X4 + 1 X15 = 18000
0 X1 + 1 X5 + 1 X9 + 1 X16 = 27000
0 X1 + 0.6 X6 -0.4 X7 -0.4 X8 + 1 X22 = 0
0 X1 -0.35 X6 + 0.65 X7 -0.35 X8 + 1 X21 = 0
0 X1 -0.25 X6 -0.25 X7 + 0.75 X8 + 1 X20 = 0
0 X1 + 0.5 X9 -0.5 X10 -0.5 X11 + 1 X17 = 0
0 X1 -0.2 X9 + 0.8 X10 -0.2 X11 + 1 X18 = 0
0 X1 -0.1 X9 -0.1 X10 + 0.9 X11 -1 X19 + 1 X23 = 0

X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22, X23 ≥ 0

Fig. 10 Model in www.PHPSimplex.com.

6.2 Memetic Algorithm or Hybrid Method that combines Genetic Algorithm and Heuristics

The prototype has the following features:

- Random generation of the initial population.
- Number of evaluated individuals = 1000.
- Mutation probabilities = 20 %.
- Population size = 20.
- Uniform crossover (Syswerda, 1989).
- The Fitness Value is the maximum value of z (benefit).
- Roulette selection (Michalewicz, 1996).
- Worst Among Most Similar Replacement (WAMS) (Shuhei, 2003). Replacement based on the euclidean distance between two individuals in order to maintain diversity in the population. If two individuals are similar (Euclidean distance shorter than 0.01 units), only the one with higher fitness value will exist in the population. Replace Worst Strategy (RW) for Individuals that satisfy the minimum distance. The worst individual in the population will be replaced, mantaining the population size.
- Local search heuristic that modifies the best individual of the population in each generation changing the real values to integer values.

The prototype follows these steps:

1. Initialization or initial population generation.
2. Fitness function computation for each individual.
Repeat
3. Application of selection operator (Roulette) to obtain two parents.
4. Application of crossover and mutation operators.
5. Application of the heuristic that searches Integer Values.
6. Fitness function computation for the obtained offspring.
7. WAMS and RW replacement.
Until stop criterion is reached.

Figure 11 shows the randomly generated initial population. The first eleven columns correspond to the 11 variables N, P, L, T, M, Gn, Gl, Gp, Hm, Hp, Hl. The last column represents the fitness value or benefit.

Figure 12 shows the evolution of the best solution to reach the optimal solution. As in Figure 11, the first eleven columns correspond to the 11 variables N, P, L, T, M, Gn, Gl, Gp, Hm, Hp, Hl. The last column, which is in the following line, represents the associated fitness value or benefit.

At the end it can be seen the final solution, which is the same as obtained using the Solver tool.

```
Ind.1:12.5  13.96  16.68  13.38  6  0  0  0  15.64  6.44  18.66  3395.74
Ind.2:18.3  15.38  0.16  6.3  19.54  0  0  0  18.14  2.22  19.38  3188.42
Ind.3:1.5  13.94  19.26  17.6  7.62  0  0  0  13.8  5.16  12.3  3078.2
Ind.4:16.04  11.56  11.72  13.86  15.14  0  0  0  3.32  1.4  18.1  3054.78
Ind.5:1.36  14.08  11.62  19  19.24  0  0  0  1.56  2.46  16.4  2874.24
Ind.6:6.06  15.74  5.58  14.4  7.06  0  0  0  16.88  0  17.94  2836.36
Ind.7:4.66  15.28  11.68  15.78  0.32  0  0  0  12.64  3.76  17.24  2751
Ind.8:13.04  9.8  12.82  16.56  2.98  0  0  0  8.6  0.24  11.12  2647.08
Ind.9:9.62  3.14  15.78  18.68  5.28  0  0  0  3.28  1.62  19.54  2612.48
Ind.10:11.56  13.1  0  14.18  3.3  0  0  0  17.36  6.74  12.14  2550.4
Ind.11:11.34  12.52  4.52  17.24  10.32  0  0  0  4.36  1.12  12.18  2528.96
Ind.12:14.14  10.92  13.68  16.42  0.86  0  0  0  0.46  0.48  11.28  2439.7
Ind.13:1.48  19.78  11.4  2.06  9.26  0  0  0  2.22  3.7  17.44  2168.26
Ind.14:12.34  19.72  4.46  2.12  7.08  0  0  0  4.02  2.9  9.82  2070
Ind.15:1.8  9.68  6.82  2.32  12.28  0  0  0  14.02  1.36  16.7  2051.64
Ind.16:0.22  12.26  6.14  7.48  16.1  0  0  0  4.64  0.4  14.56  2021.14
Ind.17:8.36  11.84  15.02  7.46  2.14  0  0  0  1.74  1.48  9  1984.1
Ind.18:11.26  11.24  4.38  2.28  8.5  0  0  0  9.14  1.78  11.88  1959.64
Ind.19:6.24  9.74  12.14  0.88  0.1  0  0  0  13.78  6.1  13.14  1951.72
Ind.20:13.66  4.1  18.48  1.14  0.88  0  0  0  2.24  2.54  7.92  1708.46
```

Figure 11 Randomly generated initial population

```
28.584  19.8013  45.9144  23.584  21.6963  0  0  0  21.8657  6.1272  29.2974
6618.25

83.46  60.2009  226.126  73.1335  46.3958  0  0  0  51.9172  22.3056  41.8933
21001.8

2001.61  1758.02  20817.6  1807.27  1375.87  0  0  0  301.993  120.43  182.029

1.03725e+006

3906.23  3507.7  20773.9  3698.4  2809.55  0  0  0  375.487  149.632  226.046
1.29182e+006

5858.55  5317.67  20727.3  5587.73  4220.46  0  0  0  453.202  181.225  272.633

1.54974e+006

7777.95  7036.65  20668.2  7481.65  5701.01  0  0  0  551.933  220.396  331.719

1.8059e+006

15525.2  14197.8  20449.1  15053.2  11487  0  0  0  915.372  366.348  550.903
2.83875e+006

17470.1  15986.3  20398.3  16909.8  12893.7  0  0  0  998.954  399.873  600.739

3.09428e+006

19358.2  17815.4  20361.4  17999.9  14392  0  0  0  1061.92  424.505  637.877
3.31998e+006
24945.6  23329.7  20240.2  18000  18983.7  0  0  0  1264.78  505.881  759.178
3.86459e+006

26885  24478.5  20207.6  18000  20528.3  0  0  0  1312.86  521.498  791.771
4.02233e+006

28915.2  24478.9  20185.3  17999.9  22254.8  0  0  0  1335.47  521.038  814.588

4.14398e+006

30623.5  24483.8  20174  17999.4  23656  0  0  0  1340.67  515.457  825.685
4.24454e+006

31999.9  24488  20167.9  17998.3  24773.1  0  0  0  1341.67  511.957  832.084
4.32516e+006

31999.9  24479.3  20151  18000  25630.6  0  0  0  1368.67  520.656  848.984
4.35057e+006
```

31999.9	24481.3	20153	18000	25634.6	0	0	0	1364.67	518.656	846.984	4.35063e+006
31999.9	24481.3	20153	18000	25634.6	0	0	0	1364.67	518.656	846.984	4.35063e+006
31999.9	24516.3	20168	18000	25684.6	0	0	0	1314.67	483.656	831.984	4.35151e+006
31999.9	24549.3	20182	18000	25731.6	0	0	0	1267.67	450.656	817.984	4.35233e+006
31999.9	24598.3	20200	18000	25798.8	0	0	0	1200.67	401.656	799.984	4.35354e+006
31999.9	24634.3	20215	18000	25849.8	0	0	0	1149.67	365.656	784.984	4.35444e+006
31999.9	24691.3	20234	18000	25925.9	0	0	0	1073.67	308.656	765.984	4.35583e+006
31999.9	24729.3	20243	18000	25973.1	0	0	0	1026.67	270.656	756.984	4.35673e+006
31999.9	24773.3	20256	18000	26030.2	0	0	0	969.67	226.656	743.984	4.35779e+006
31999.9	24806.3	20267	18000	26074.2	0	0	0	925.67	193.656	732.984	4.3586e+006
31999.9	24919.3	20303	18000	26223.3	0	0	0	776.67	80.6563	696.984	4.36134e+006
31999.9	24983.3	20317	18000	26301.3	0	0	0	698.67	16.6563	682.984	4.36284e+006
31999.9	24999.3	20335	18000	26335.3	0	0	0	664.67	0.65625	664.984	4.36332e+006
31999.9	24999.3	20366	18000	26366.3	0	0	0	633.67	0.65625	633.984	4.36354e+006
31999.9	24999.3	20387	18000	26387.3	0	0	0	612.67	0.65625	612.984	4.36369e+006
31999.9	24999.3	20509	18000	26509.3	0	0	0	490.67	0.65625	490.984	4.36454e+006
31999.9	24999.3	20531	18000	26531.3	0	0	0	468.67	0.65625	468.984	4.36469e+006
31999.9	24999.3	20557	18000	26557.3	0	0	0	442.67	0.65625	442.984	4.36488e+006
31999.9	24999.3	20586	18000	26586.3	0	0	0	413.67	0.65625	413.984	4.36508e+006
31999.9	24999.3	20509	18000	26509.3	0	0	0	490.67	0.65625	490.984	4.36454e+006
31999.9	24999.3	20531	18000	26531.3	0	0	0	468.67	0.65625	468.984	4.36469e+006
31999.9	24999.3	20557	18000	26557.3	0	0	0	442.67	0.65625	442.984	4.36488e+006
31999.9	24999.3	20586	18000	26586.3	0	0	0	413.67	0.65625	413.984	4.36508e+006

```
31999.9  24999.3  20809  18000  26809.3  0  0  0  190.67  0.65625  190.984
4.36664e+006

31999.9  24999.3  20829  18000  26829.3  0  0  0  170.67  0.65625  170.984
4.36678e+006

31999.9  24999.3  20849  18000  26849.3  0  0  0  150.67  0.65625  150.984
4.36692e+006

31999.9  24999.3  20870  18000  26870.3  0  0  0  129.67  0.65625  129.984
4.36707e+006

31999.9  24999.3  20896  18000  26896.3  0  0  0  103.67  0.65625  103.984
4.36725e+006

31999.9  24999.3  20920  18000  26920.3  0  0  0  79.6702  0.65625  79.9837
4.36742e+006

31999.9  24999.3  20944  18000  26944.3  0  0  0  55.6702  0.65625  55.9837
4.36758e+006

31999.9  24999.3  20959  18000  26959.3  0  0  0  40.6702  0.65625  40.9837
4.36769e+006

32000  25000  20992  18000  26992  0  0  0  8  0  8
4.36794e+006

32000  25000  21000  18000  27000  0  0  0  0  0  0
4.368e+006
```

Figure 12 Evolution of the best individual to reach the optimal solution

6.3 Solver tool of Microsoft Excel 2007

Solver tool of Microsoft Excel 2007 has been used to find the solution of "The Juice Processing Problem", whose model appears in section 5.2 of this article.

The result has been very successful, reaching exactly the same solution proposed by the memetic algorithm, so it can be said that both the proposed Memetic algorithm and Microsoft Excel Solver tool reach optimal solutions for Operational Research problems, including high complex problems as the one proposed in this paper.

Conclusions

There are high complex problems, like the one presented in this article, for which traditional optimization techniques, inverse matrix, do not get the best results. Simplex algorithm may be infeasible in large problems and it can find difficulties to solve problems with equality and inequality constraints using the two phases and penalties methods. Finally, it is presented a self-created prototype that corresponds to a Memetic algorithm or hybrid algorithm based on Genetic algorithms and a local search heuristic technique that provides to the prototype, the ability to reach optimal solutions even in problems with integer solutions through the evolution of these solutions.

REFERENCES

1. Chica, M., Cordón, O., Damas, S., et al., (2009). *Heurísticas constructivas multiobjetivo para el problema de equilibrado de líneas de montaje considerando tiempo y espacio*, Comunicación, MAEB'09, VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Málaga, pp. 649-656
2. Cotta, C., (2007). *Una Visión General de los Algoritmos Meméticos*. 1ª Edición. Procedimientos Metaheurísticos en Economía y Empresa. España, ASEPUMA., pp. 139-166
3. Darwin, C., (1859). *On the Origin of Species by Means of Natural Selection*, London, Murray, pp. 502
4. Davis, L., (1991). *Handbook of Genetic Algorithms*, 1st Edition. New York, Van Nostrand Reinhold, pp. 385
5. Holland, J., (1975). (*Adaptation in Natural and Artificial Systems*, 1st Edition, USA, Ann Arbor, University of Michigan Press, pp.228
6. Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. 1st Edition. USA, Springer, pp. 387
7. Michalewicz, Z, Fogel, D. B., (2004). *How to Solve It: Modern Heuristics*. 2nd Edition Revised and Extended. USA, Springer, pp.467
8. Moscato, P., (1989). *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetics Algorithms*. Technical Report Caltech Concurrent Computation Program, C3P. California Institute of Technology, Pasadena. California, USA, Report 826
9. Oduguwa, V., Tiwari, A., Roy, R. (2005). Evolutionary computing in manufacturing industry: an overview of recent applications. *Applied Soft Computing*, pp. 281–299
10. Pearl, J., (1984). Heuristics. *Intelligent Search Strategies for Computer Problem Solving.* 1st Edition. Massachusests: Addison Wesley, pp 399;
11. Pearl, J., (1983). On the Discovery and Generation of Certain Heuristics. *AI Magazine* Volume 4 Number 1, 1983
12. Ríos, S., (1996). *Investigación Operativa. Programación lineal y aplicaciones*. 1ª Edición. Madrid, Editorial Centro de Estudios Ramón Areces S.A.
13. Syswerda, G., (1989). *Uniform crossover in genetic algorithms*. Proceedings of the 3rd International Conference on Genetic Algorithms, USA, pp.2-9
14. Shuhei, K. (2003). A Genetic Algorithm with Distance Independent Diversity Control for High Dimensional Function Optimization. *Transactions of the Japanese Society for Artificial Intelligence*, Volume 18, pp. 193-202
15. Whitley, D. (1994). A Genetic Algorithm Tutorial, *Statistics and Computing*, Volume 4, Issue 2, pp. 65-85