

Algoritmos Meméticos en problemas de Investigación Operativa

Memetic Algorithms in Operational Research problems

**Jesús M. Larrañaga Lesaca¹, Ekaitz Zulueta Guerrero², Fernando Elizagarate
Ubis³, Jon Alzola Bernardo⁴**

¹ Dpto. de Organización de Empresas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. Universidad del País Vasco Euskal Herriko Unibertsitatea (UPV-EHU). Nieves Cano 12, 01006 Vitoria-Gasteiz.

² Dpto. de Ingeniería de sistemas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. Universidad del País Vasco Euskal Herriko Unibertsitatea (UPV-EHU). Nieves Cano 12, 01006 Vitoria-Gasteiz.

³ Dpto. de Organización de Empresas. Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. Universidad del País Vasco Euskal Herriko Unibertsitatea (UPV-EHU). Nieves Cano 12, 01006 Vitoria-Gasteiz.

⁴ Investigador predoctoral

Resumen.

En este artículo se plantea la resolución de un problema de Investigación Operativa utilizando PHPSimplex (herramienta online de resolución de problemas de optimización utilizando el método Simplex), Solver de Microsoft Excel y un prototipo híbrido que combina las teorías de los Algoritmos Genéticos con una técnica heurística de búsqueda local. La hibridación de estas dos técnicas es conocida como Algoritmo Memético. Este prototipo será capaz de resolver problemas de Optimización con función de maximización o minimización conocida, superando las restricciones que se planteen.

Los tres métodos conseguirán buenos resultados ante problemas de Investigación Operativa.

El objetivo, además de resolver el problema propuesto, es comparar cómo se comportan los tres métodos anteriormente citados ante el problema y cómo afrontan las dificultades que éste presenta. Además, este artículo pretende dar a conocer diferentes técnicas de apoyo a la toma de decisiones, con la intención de que se utilicen cada vez más en el entorno empresarial sustentando, de esta manera, las decisiones mediante la matemática o la Inteligencia Artificial y no basándose únicamente en la experiencia.

Abstract.

This article tries to give solution to an Operational Research problem by using PHPSimplex (online tool to solve optimization problems by using Simplex method), Microsoft Excel Solver and a hybrid prototype which combines the theories of Genetic Algorithms with a local search heuristic technique. Hybridization of these two techniques is known as Memetic Algorithm. This prototype will be able to solve optimization problems with known maximization or minimization function, overcoming the arised constraints.

The three methods will get good results with problems of Operational Research.

The goal, besides solving the proposed problem, is to compare how the three methods mentioned behave with the problem and how they face the difficulties it presents. Additionally, this article tries to present different techniques to support 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.

Palabras clave: Simplex, Solver, Algoritmos Genéticos, Investigación Operativa, Programación Lineal, Heurísticas, Método Híbrido, Algoritmos Meméticos, Métodos cuantitativos, Optimización, Restricciones.

Keywords: Simplex, Solver, Genetic Algorithms, Operational Research, Linear Programming, Heuristics, Hybrid Methods, Memetic Algorithms, Quantitative Methods, Optimization, Constraints.

1. Introducción

La Investigación Operativa es una moderna disciplina científica que se caracteriza por la aplicación de teoría, métodos y técnicas especiales, para buscar la solución de los problemas. Para hallar la solución, la Investigación Operativa generalmente representa el problema como un modelo matemático, que es analizado y evaluado previamente. Es imprescindible por tanto tener la información suficiente como para elaborar un modelo

fiel a la realidad, en caso contrario las decisiones se tomarían a través de la experiencia o se establecería el modelo a través de la simulación de los procesos productivos (Villanueva, 2008).

El objetivo más importante de la aplicación de la Investigación Operativa es apoyar en la “toma de decisiones optimas”.

En este artículo se plantea la resolución de un problema conocido como el “Problema de Elaboración de zumos”. El enunciado de este problema aparece en los libros: Investigación Operativa. Programación lineal y aplicaciones de Sixto Ríos (1996), Problemas de Investigación Operativa, Sixto Ríos (Ra-Ma 2006); aunque planteado aparece sin resolver.

“El Problema de Elaboración de zumos”, por tanto, no tiene solución conocida, por lo que se buscará la resolución con métodos clásicos de optimización, como pueden ser los métodos cuantitativos, particularmente el método Simplex a través de la herramienta gratuita www.PHPSimplex.com, y utilizando la herramienta Solver de Microsoft Excel. Por último, se mostrará cómo un prototipo, desarrollado en lenguaje de programación C++ y basado en la metaheurística conocida como Algoritmos Genéticos, junto a una heurística creada para mejorar el funcionamiento, es capaz de llegar a la solución óptima. Esta hibridación de ambas técnicas recibe el nombre de Algoritmo Memético.

Antes de enfrentarse al “Problema de Elaboración de zumos”, se probará el correcto funcionamiento de las tres técnicas en un problema más sencillo, el del “Productor de Cerveza” (Sixto Ríos, 1996).

2. Técnicas de optimización

Existen varias alternativas para dar solución a un problema complejo que cumpla las restricciones propuestas y optimice, maximizando o minimizando, una función conocida. En este artículo utilizaremos tres, los cuales se describen a continuación:

2.1. PHPSimplex

La web www.PHPSimplex.com permite resolver problemas de optimización utilizando el método Simplex.

El método Simplex, descubierto por el matemático norteamericano George Bernard Dantzig en 1947, es una técnica popular para dar soluciones numéricas a problemas de programación lineal en los que intervienen tres o más variables.

El álgebra matricial y el proceso de eliminación de Gauss-Jordan para resolver un sistema de ecuaciones lineales constituyen la base del método simplex. La resolución de programas lineales mediante el método Simplex implica la realización de gran cantidad de cálculos a través de sucesivas tablas, sobre todo cuando el número de variables y/o restricciones es relativamente elevado. En casos reales, la magnitud de los problemas hace necesario recurrir al uso del computador.

PHPSimplex permite hacer la resolución paso a paso viendo los cambios que se producen en las tablas del Simplex.

2.2. Herramienta Solver de Microsoft Excel

Solver es una herramienta para resolver y optimizar ecuaciones mediante el uso de métodos numéricos. Solver se puede utilizar para optimizar funciones de una o más

variables, sin o con restricciones. La opción Solver de EXCEL sirve para resolver problemas de optimización lineal y no lineal; también se pueden indicar restricciones enteras sobre las variables de decisión. Con Solver es posible resolver problemas que tengan hasta 200 variables de decisión, 100 restricciones explícitas y 400 simples (cotas superior e inferior o restricciones enteras sobre las variables de decisión). Para acceder a Solver, seleccione “Tools” en el menú principal y luego Solver.

La forma de usar la herramienta Solver se describe a continuación. La ventana Solver Parameters se utiliza para describir el problema de optimización a EXCEL. El campo Set Target Cell contiene la celda donde se encuentra la función objetivo correspondiente al problema en cuestión. Si desea hallar el máximo o el mínimo, seleccione Max o Min. Si la casilla Value of está seleccionada, Solver tratará de hallar un valor de la celda igual al valor del campo que se encuentra a la derecha de la selección. El cuadro de diálogo By Changing Cells contendrá la ubicación de las variables de decisión para el problema. Por último, las restricciones se deben especificar en el campo Subject to the Constraints haciendo clic en Add. El botón Change permite modificar las restricciones recién introducidas y Delete sirve para borrar las restricciones precedentes. Reset All borra el problema en curso y restablece todos los parámetros a sus valores por defecto.

2.3. Algoritmos Genéticos, Heurísticas y Algoritmos Meméticos

En los años 70, y basándose en la Teoría de la Evolución (Darwin, 1859), se desarrolla una novedosa técnica de búsqueda conocida como Algoritmos Genéticos (Holland, 1975), la cual, va seleccionando las mejores soluciones posibles hasta llegar a la solución óptima a través de distintos métodos basados en la naturaleza, como son la selección, el cruzamiento o la mutación. De esta manera se produce una evolución en las soluciones hacia la solución óptima. Los principios básicos de los Algoritmos Genéticos se encuentran bien descritos en numerosos textos, (Davis, 1991), (Michalewicz, 1996), (Whitley, 1994).

Las heurísticas son técnicas basadas en la experiencia que ayudan a resolver un problema. Son reglas, que surgen de la experiencia, las cuales se utilizan cuando se quiere obtener una buena solución en un tiempo razonable o cuando no existe un método capaz de alcanzar soluciones óptimas, bajo las restricciones del problema. Puede encontrarse diversa información en distintos artículos, (Michalewicz and Fogel, 2004), (Pearl, 1984), (Chica, et al., 2009).

Los Algoritmos Meméticos surgen como combinación de los Algoritmos Genéticos y técnicas heurísticas, normalmente de búsqueda local. Los individuos creados por ambas técnicas compiten y cooperan de manera sinérgica (Moscato, 1989), obteniendo muy buenos resultados. Aparecen bien descritos en el siguiente artículo, (Cotta, 2007).

3. El “Problema del Productor de Cerveza”

3.1. Enunciado

Una fábrica de cervezas produce tres tipos distintos denominados negra, rubia y de baja graduación. Para su obtención son necesarios, además de agua y lúpulo para los cuales no hay limitación de disponibilidad, malta y levadura, que limitan la capacidad diaria de producción. La tabla siguiente da la cantidad necesaria de cada uno de estos recursos para producir un litro de cada una de las respectivas cervezas, los kilos disponibles de cada recurso y el beneficio en unidades monetarias (u.m.) por litro de cada cerveza

producida. El problema del fabricante consiste en decidir cuánto debe fabricar de cada cerveza para que el beneficio total diario sea máximo.

Tabla 1: Resumen del enunciado del problema del productor de cerveza

	Negra	Rubia	Baja graduación	Disponibilidad
Malta	2	1	2	30
Levadura	1	2	2	45
Beneficio	4	7	3	

3.2. Modelo del problema

Variables de decisión:

- X_1 = Producción en litros de cerveza negra (N) por día.
- X_2 = Producción en litros de cerveza rubia (R) por día.
- X_3 = Producción en litros de cerveza de baja graduación (B) por día.

Restricciones:

- $2X_1 + X_2 + 2X_3 \leq 30$
- $X_1 + 2X_2 + 2X_3 \leq 45$

Función de maximización del beneficio:

$$\text{Max } z = 4X_1 + 7X_2 + 3X_3$$

4. Resolución del “Problema del Productor de Cerveza”

4.1. PHPSimplex

Como se ha comentado anteriormente, se va a utilizar el programa gratuito integrado en la web www.PHPSimplex.com para buscar las soluciones, y es que PHPSimplex es una herramienta online para resolver problemas de programación lineal cuyo uso es libre y gratuito.

Introducimos el modelo del problema para ver los resultados y comprobar si llega a la misma solución que hemos obtenido anteriormente con el Solver y el prototipo de Algoritmos Genéticos.

En primer lugar, elegimos el método Simplex, e introducimos el número de variables y restricciones, tres y dos respectivamente, para este problema, y pulsamos sobre el botón de “Continuar”. Ver Figura 1.

PHPSimplex

teoría Ejemplo Ayuda Salir

Método: Simplex/Dos Fases

¿Cuántas variables de decisión tiene el problema? 3

¿Cuántas restricciones? 2

Continuar

Fig. 1: Primera fase para la resolución del problema con PHPSimplex.

A continuación, elegimos la opción de Maximizar, ya que buscamos el beneficio máximo para este problema y vamos rellenando los huecos con los respectivos coeficientes, una vez rellenados los campos pulsamos el botón de “Continuar”. Ver Figura 2.

¿Cual es el objetivo de la funcion? Maximizar

Funcion: 4 X1 + 7 X2 + 3 X3

Restricciones:

2 X1 + 1 X2 + 2 X3 ≤ 30

1 X1 + 2 X2 + 2 X3 ≤ 45

Continuar

Fig. 2: Segunda fase para la resolución del problema con PHPSimplex.

Una vez introducido el modelo, el programa nos lo muestra ordenadamente, como podemos observar en la Figura 3.

Pasamos el problema a la forma estandar, añadiendo variables de exceso, holgura, y artificiales segun corresponda.

<p>MAXIMIZAR: $4 X_1 + 7 X_2 + 3 X_3$</p> <p>$2 X_1 + 1 X_2 + 2 X_3 \leq 30$</p> <p>$1 X_1 + 2 X_2 + 2 X_3 \leq 45$</p> <p>$X_1, X_2, X_3 \geq 0$</p>		<p>MAXIMIZAR: $4 X_1 + 7 X_2 + 3 X_3 + 0 X_4 + 0 X_5$</p> <p>$2 X_1 + 1 X_2 + 2 X_3 + 1 X_4 = 30$</p> <p>$1 X_1 + 2 X_2 + 2 X_3 + 1 X_5 = 45$</p> <p>$X_1, X_2, X_3, X_4, X_5 \geq 0$</p>
---	--	---

Pasamos a construir la primera tabla del método del Simplex.

Continuar

Solución directa

Fig. 3: Tercera fase para la resolución del problema con PHPSimplex.

Una vez comprobado que todos los datos son correctos pulsamos sobre el botón “Solución directa” para que nos de la decisión óptima que maximiza el beneficio para los datos introducidos. Ver Figura 4.

La solución óptima es $Z = 160$
 $X_1 = 5$
 $X_2 = 20$
 $X_3 = 0$

Fig. 4: Cuarta fase para la resolución del problema con PHPSimplex.

4.2. Herramienta Solver de Excel

En primer lugar se introducen los datos del modelo del problema en Excel, para después aplicar la herramienta Solver. El resultado, como podemos observar en la Figura 5, es el mismo que habíamos obtenido anteriormente para este problema utilizando PHPSimplex.

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

Fig. 5: Modelo del problema en Excel y resultado obtenido a través del Solver.

4.3. Algoritmo Memético o hibridación entre Algoritmos Genéticos y Técnicas Heurísticas

Para la realización del prototipo basado en Algoritmos Genéticos se han hecho numerosas pruebas definiendo distintos tipos de mutación, cruzamiento, tamaño de población, reemplazamiento... Tras estas pruebas se han seleccionado los métodos y valores con los que se han obtenido los mejores resultados. Sin embargo, se observa que el funcionamiento del prototipo es excelente, excepto en soluciones que requieren valores enteros, ya que el prototipo trabaja con números reales y le cuesta alcanzar las soluciones óptimas enteras. Por ello, se ha implementado una técnica heurística de búsqueda local que modifica la mejor solución cambiando los valores reales de las variables por los enteros más cercanos, creando así nuevos individuos que también compiten por entrar en la población. Los resultados del prototipo híbrido no han podido ser más satisfactorios, como se verá a continuación en la Figura 6.

Las características del algoritmo son las siguientes:

- Generación aleatoria de la población inicial.
- Número de individuos evaluados = 1000.
- Probabilidad de mutación = 20 %.
- Tamaño de la población = 20.
- Cruzamiento uniforme (cada variable o rasgo genético de un nuevo individuo tiene el 50% de posibilidades de pertenecer a un progenitor o al otro), (Syswerda, 1989).
- El valor de fitness es el máximo valor de z (beneficio).
- Selección por Ruleta (Michalewicz, 1996).

- Reemplazo por distancia para mantener la diversidad de la población en la que dos individuos muy parecidos (distancia euclídea entre ambos menor que 0.01 uds.) no pueden coexistir en la población (Worst Among Most Similar Replacement, WAMS) (Shuhei, 2003), y mantenimiento del tamaño de la población reemplazando al peor individuo (Replace Worst Strategy, RW) siempre y cuando cumpla el criterio de diversidad anterior.
- Método heurístico que prueba los enteros más cercanos a la mejor solución de la población en cada generación.

Los pasos de ejecución del prototipo son los siguientes:

1. Generación de la población inicial a partir de la solución generada manualmente.
2. Cálculo del valor de fitness para cada individuo.

Repetir

3. Aplicación del operador de selección (Ruleta) para obtener los dos padres.
4. Aplicación de operadores de cruzamiento y mutación.
5. Aplicación del método heurístico de probar enteros.
6. Cálculo del valor de fitness para las nuevas generaciones creadas.
7. Reemplazamiento WAMS y RW.

Hasta alcanzar el criterio de parada.

La evolución de las soluciones hasta alcanzar la solución final se muestra a continuación.

En la Figura 6, puede observarse la población inicial generada aleatoriamente para solucionar el problema en estudio. La tres primeras columnas corresponden a X1, X2 y X3, respectivamente, y la cuarta columna al valor de adaptación o Fitness Value extraído de la función de maximización.

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

Fig. 6: Población inicial generada aleatoriamente.

A continuación puede observarse, en la Figura 7, la evolución de la mejor de las soluciones hasta alcanzar la decisión óptima para este problema. Puede observarse como el método heurístico de ajuste ha funcionado correctamente entre la Generación 19 y la 20, acelerando enormemente el proceso de búsqueda de soluciones óptimas.

Mejor Individuo	Generación 1:	6	11	3	110
Mejor Individuo	Generación 2:	6.1251	13.1777	2.1	123.044
Mejor Individuo	Generación 3:	6.2511	15.2777	1.1	135.248
Mejor Individuo	Generación 4:	6.3611	15.2777	0.99	135.358
Mejor Individuo	Generación 5:	5.4705	16.3927	1.09	139.901
Mejor Individuo	Generación 6:	5.6621	16.4927	1.09	141.367
Mejor Individuo	Generación 7:	4.7621	17.6181	1.2669	146.176
Mejor Individuo	Generación 8:	4.8621	17.7181	1.2669	147.276
Mejor Individuo	Generación 9:	4.8621	17.7181	1.2669	147.276
Mejor Individuo	Generación 10:	4.9621	19.7181	0.1669	158.376
Mejor Individuo	Generación 11:	4.9649	19.7181	0.1669	158.387
Mejor Individuo	Generación 12:	4.9649	19.7181	0.1669	158.387
Mejor Individuo	Generación 13:	4.9649	19.7181	0.1669	158.387
Mejor Individuo	Generación 14:	4.9649	19.7181	0.1669	158.387
Mejor Individuo	Generación 15:	4.9649	19.7181	0.1669	158.387
Mejor Individuo	Generación 16:	4.9649	19.7181	0.1728	158.405
Mejor Individuo	Generación 17:	4.9649	19.7181	0.1728	158.405
Mejor Individuo	Generación 18:	4.9254	19.8393	0.1396	158.995
Mejor Individuo	Generación 19:	4.9254	19.8393	0.1396	158.995
Mejor Individuo	Generación 20:	5	20	0	160

Fig. 7: Evolución del mejor individuo hacia la solución óptima.

En la Figura 8, puede observarse las 20 soluciones que conforman en este caso la población final, encabezada por la mejor solución encontrada en la primera posición.

Poblacion final alcanzada:

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

Fig. 8: Población final alcanzada.

Por último, el programa emite la recomendación final, correspondiente al mejor individuo o solución encontrada (Figura 9), la cual corresponde a $X_1 = 5$, $X_2 = 20$, $X_3 = 0$, lo que nos da un beneficio máximo de 160.

La decision final recomendada es: 5 20 0 160

Fig. 9: Solución final recomendada.

Puede observarse el correcto funcionamiento de las tres técnicas de optimización, ya que todas llegan a alcanzar la solución óptima para este problema.

5. El “Problema de Elaboración de zumos”

5.1. Enunciado

Una empresa de alimentación produce zumos de pera, naranja, limón, tomate, manzana, además de otros dos tipos denominados H y G que son combinados de algunos de los anteriores. La disponibilidad de fruta para el período próximo, así como los costes de producción y precios de venta para los zumos, vienen dados en la Tabla 1.

Tabla 1. Especificaciones de los zumos de fruta simples.

Fruta	Disponibilidad máxima (Kg)	Coste (cénts de €/Kg)	Precio de venta (cénts de €/Kg)
Naranja (N)	32.000	94	129
Pera (P)	25.000	87	125
Limón (L)	21.000	73	110
Tomate (T)	18.000	47	88
Manzanas (M)	27.000	68	97

Las especificaciones y precios de venta de los combinados vienen dados en la Tabla 2.

Tabla 2. Especificaciones de los zumos de fruta combinados.

Combinado	Especificación	Precio de venta (céntos de €/Kg)
H	No más del 50% de M No más del 20% de P No menos del 10% de L	100
G	40% de N 35% de L 25% de P	120

La demanda de los distintos zumos es grande por lo que se espera vender toda la producción. Por cada Kg de fruta se produce un litro del correspondiente zumo. Formular un programa lineal para determinar los niveles de producción de los siete zumos de manera que se tenga beneficio máximo para el período entrante.

5.2. Modelo del problema

Se ha reducido el número de restricciones de 13 a 11, desapareciendo los zumos combinados, H y G, como variable y apareciendo únicamente como variable la cantidad de cada zumo simple que forman un combinado, $H = H_m + H_p + H_l$ y $G = G_n + G_l + G_p$.

Las variables de decisión, por tanto, son: N (Naranja), P (Pera), L (Limón), T (Tomate), M (Manzana), G_n (Parte de Naranja del combinado G), G_l (Parte de Limón del combinado G), G_p (Parte de Pera del combinado G), H_m (Parte de Manzana del combinado H), H_p (Parte de Pera del combinado H), H_l (Parte de Limón del combinado H).

Las Restricciones son las siguientes:

$$N + G_n \leq 32000$$

$$P + H_p + G_p \leq 25000$$

$$L + H_l + G_l \leq 21000$$

$$T \leq 18000$$

$$M + H_m \leq 27000$$

$$G_n = 0.4(G_n + G_l + G_p)$$

$$G_l = 0.35(G_n + G_l + G_p)$$

$$G_p = 0.25(G_n + G_l + G_p)$$

$$H_m \leq 0.5(H_m + H_p + H_l)$$

$$H_p \leq 0.2(H_m + H_p + H_l)$$

$$H_l \geq 0.1(H_m + H_p + H_l)$$

La función de maximización del beneficio es:

$$\text{Max } z = 35N + 38P + 37L + 41T + 29M + 120(G_n + G_l + G_p) - 94G_n - 73G_l - 87G_p + 100(H_m + H_p + H_l) - 68H_m - 87H_p - 73H_l$$

Simplificando:

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

6. Resolución del “Problema de Elaboración de zumos”

6.1. Método Simplex

Se ha introducido el modelo en la web www.PHPSimplex.com como puede observarse en la Figura 10.

The screenshot shows the PHPSimplex website interface. At the top, the title "PHPSimplex" is displayed. Below it, the "Método" is set to "Simplex / Dos Fases". The user has entered "11" for the number of decision variables and "11" for the number of constraints. A "Continuar" button is visible.

Below this, the objective function is defined as "Maximizar" with the coefficients: 35, 38, 37, 41, 29, 26, 47, 33, 32, 13, 27. The objective function is expressed as: $35 X_1 + 38 X_2 + 37 X_3 + 41 X_4 + 29 X_5 + 26 X_6 + 47 X_7 + 33 X_8 + 32 X_9 + 13 X_{10} + 27 X_{11}$.

The constraints are listed as follows:

- 1 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 1 X_6 + 0 X_7 + 0 X_8 + 0 X_9 + 0 X_{10} + 0 X_{11} \leq 32000$
- 0 $X_1 + 1 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 0 X_6 + 0 X_7 + 1 X_8 + 0 X_9 + 1 X_{10} + 0 X_{11} \leq 25000$
- 0 $X_1 + 0 X_2 + 1 X_3 + 0 X_4 + 0 X_5 + 0 X_6 + 1 X_7 + 0 X_8 + 0 X_9 + 1 X_{10} + 1 X_{11} \leq 21000$
- 0 $X_1 + 0 X_2 + 0 X_3 + 1 X_4 + 0 X_5 + 0 X_6 + 0 X_7 + 0 X_8 + 0 X_9 + 0 X_{10} + 0 X_{11} \leq 18000$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 1 X_5 + 0 X_6 + 0 X_7 + 0 X_8 + 1 X_9 + 0 X_{10} + 0 X_{11} \leq 27000$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 0,6 X_6 + -0,4 X_7 + -0,4 X_8 + 0 X_9 + 0 X_{10} + 0 X_{11} = 0$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + -0,25 X_6 + 0,65 X_7 + -0,35 X_8 + 0 X_9 + 0 X_{10} + 0 X_{11} = 0$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + -0,25 X_6 + -0,25 X_7 + 0,75 X_8 + 0 X_9 + 0 X_{10} + 0 X_{11} = 0$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 0 X_6 + 0 X_7 + 0 X_8 + 0,5 X_9 + -0,5 X_{10} + -0,5 X_{11} \leq 0$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 0 X_6 + 0 X_7 + 0 X_8 + -0,2 X_9 + 0,8 X_{10} + -0,2 X_{11} \leq 0$
- 0 $X_1 + 0 X_2 + 0 X_3 + 0 X_4 + 0 X_5 + 0 X_6 + 0 X_7 + 0 X_8 + -0,1 X_9 + -0,1 X_{10} + 0,9 X_{11} \geq 0$

A "Continuar" button is located at the bottom of the constraints section.

Figura 10. Datos introducidos para resolver el problema mediante Simplex a través de www.PHPSimplex.com.

Una vez introducidos los datos, el programa nos muestra el modelo con las variables de holgura y variables artificiales añadidas.

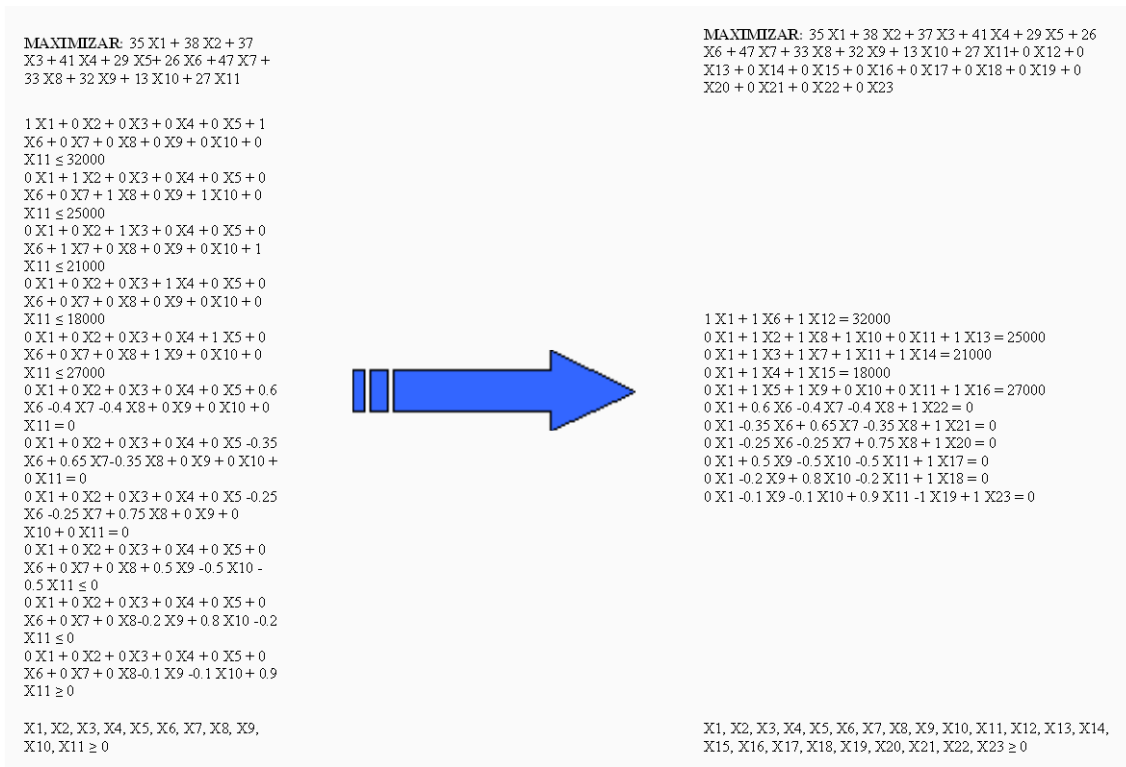


Figura 11. Modelo en www.PHPSimplex.com con los datos introducidos.

Una vez llegado a este punto, dando al botón de continuar o de Solución Directa que observamos en la Figura 3, el programa nos proporciona un resultado óptimo igual al alcanzado por el Algoritmo Memético propuesto en el punto 6.2 de este artículo.

6.2. Algoritmo Memético o hibridación entre Algoritmos Genéticos y Técnicas Heurísticas

Se ha implementado un prototipo de Algoritmo Genético con las siguientes características:

- Generación aleatoria de la población inicial.
- Número de individuos evaluados = 1000.
- Probabilidad de mutación = 20 %.
- Tamaño de la población = 20.
- Cruzamiento uniforme (cada variable o rasgo genético de un nuevo individuo tiene el 50% de posibilidades de pertenecer a un progenitor o al otro), Syswerda (1989).
- El valor de fitness es el máximo valor de z (beneficio).
- Selección por Ruleta, Michalewicz 1996.
- Reemplazo por distancia para mantener la diversidad de la población en la que dos individuos muy parecidos (distancia euclídea entre ambos menor que 0.01 uds.) no pueden coexistir en la población (Worst Among Most Similar Replacement, WAMS), Shuhe 2003, y mantenimiento del tamaño de la población reemplazando

al peor individuo (Replace Worst Strategy, RW) siempre y cuando cumpla el criterio de diversidad anterior.

- Método heurístico que prueba los enteros más cercanos a la mejor solución de la población en cada generación.

Los pasos de ejecución del prototipo son los siguientes:

1. Generación de la población inicial a partir de la solución generada manualmente.
2. Cálculo del valor de fitness para cada individuo.

Repetir

3. Aplicación del operador de selección (Ruleta) para obtener los dos padres.
4. Aplicación de operadores de cruzamiento y mutación.
5. Aplicación del método heurístico de probar enteros.
6. Cálculo del valor de fitness para las nuevas generaciones creadas.
7. Reemplazamiento WAMS y RW.

Hasta alcanzar el criterio de parada.

Con estas características alcanza la solución óptima para este problema con un beneficio de 4368000€

En la Figura 13, se puede observar la población inicial generada aleatoriamente. Las 11 primeras columnas, corresponden con el valor que toman las variables N, P, L, T, M, Gn, Gl, Gp, Hm, Hp, Hl. La última corresponde al beneficio que se obtendría con los valores asociados de las 11 variables.

Poblacion inicial												
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

Figura 13. Población inicial generada aleatoriamente.

En la Figura 14, puede observarse la evolución del mejor individuo de la población hasta alcanzar la solución óptima. Al igual que en la Figura 3, las 11 primeras columnas, corresponden con el valor que toman las variables N, P, L, T, M, Gn, Gl, Gp,

Hm, Hp, Hl. La última, que en esta ocasión se encuentra en la siguiente línea, corresponde al beneficio que se obtendría con los valores asociados de las 11 variables.

28.584 6618.25	19.8013	45.9144	23.584	21.6963	0	0	0	21.8657	6.1272	29.2974
83.46 21001.8	60.2009	226.126	73.1335	46.3958	0	0	0	51.9172	22.3056	41.8933
2001.61 1.03725e+006	1758.02	20817.6	1807.27	1375.87	0	0	0	301.993	120.43	182.029
3906.23 1.29182e+006	3507.7	20773.9	3698.4	2809.55	0	0	0	375.487	149.632	226.046
5858.55 1.54974e+006	5317.67	20727.3	5587.73	4220.46	0	0	0	453.202	181.225	272.633
7777.95 1.8059e+006	7036.65	20668.2	7481.65	5701.01	0	0	0	551.933	220.396	331.719
15525.2 2.83875e+006	14197.8	20449.1	15053.2	11487	0	0	0	915.372	366.348	550.903
17470.1 3.09428e+006	15986.3	20398.3	16909.8	12893.7	0	0	0	998.954	399.873	600.739
19358.2 3.31998e+006	17815.4	20361.4	17999.9	14392	0	0	0	1061.92	424.505	637.877
24945.6 3.86459e+006	23329.7	20240.2	18000	18983.7	0	0	0	1264.78	505.881	759.178
26885 4.02233e+006	24478.5	20207.6	18000	20528.3	0	0	0	1312.86	521.498	791.771
28915.2 4.14398e+006	24478.9	20185.3	17999.9	22254.8	0	0	0	1335.47	521.038	814.588
30623.5 4.24454e+006	24483.8	20174	17999.4	23656	0	0	0	1340.67	515.457	825.685
31999.9 4.32516e+006	24488	20167.9	17998.3	24773.1	0	0	0	1341.67	511.957	832.084
31999.9 4.35057e+006	24479.3	20151	18000	25630.6	0	0	0	1368.67	520.656	848.984
31999.9 4.35063e+006	24481.3	20153	18000	25634.6	0	0	0	1364.67	518.656	846.984
31999.9 4.35063e+006	24481.3	20153	18000	25634.6	0	0	0	1364.67	518.656	846.984
31999.9 4.35151e+006	24516.3	20168	18000	25684.6	0	0	0	1314.67	483.656	831.984
31999.9 4.35233e+006	24549.3	20182	18000	25731.6	0	0	0	1267.67	450.656	817.984
31999.9 4.35354e+006	24598.3	20200	18000	25798.8	0	0	0	1200.67	401.656	799.984
31999.9 4.35444e+006	24634.3	20215	18000	25849.8	0	0	0	1149.67	365.656	784.984
31999.9 4.35583e+006	24691.3	20234	18000	25925.9	0	0	0	1073.67	308.656	765.984
31999.9 4.35673e+006	24729.3	20243	18000	25973.1	0	0	0	1026.67	270.656	756.984
31999.9 4.35779e+006	24773.3	20256	18000	26030.2	0	0	0	969.67	226.656	743.984
31999.9 4.3586e+006	24806.3	20267	18000	26074.2	0	0	0	925.67	193.656	732.984

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	20614	18000	26614.3	0	0	0	385.67	0.65625	385.984
4.36527e+006										
31999.9	24999.3	20659	18000	26659.3	0	0	0	340.67	0.65625	340.984
4.36559e+006										
31999.9	24999.3	20675	18000	26675.3	0	0	0	324.67	0.65625	324.984
4.3657e+006										
31999.9	24999.3	20700	18000	26700.3	0	0	0	299.67	0.65625	299.984
4.36588e+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										

La decision final recomendada es: 32000 25000 21000 18000 27000
0 0 0 0 0 0 con un beneficio de 4.368e+006

Figura 14. Solución alcanzada por prototipo de Algoritmos Genéticos

6.3. Herramienta Solver de Microsoft Excel 2007

Se utiliza la herramienta Solver de Microsoft Excel 2007 para encontrar la solución del problema de “Elaboración de zumos” modelado en el punto 5.2 de este artículo.

El resultado ha sido muy satisfactorio, al alcanzar exactamente la misma solución que la propuesta por el Algoritmo Memético.

7. Conclusiones

Se presentan en este artículo tres maneras de resolver problemas de Investigación Operativa, concretamente de Programación Lineal. PHPSimplex y Solver de Microsoft Excel utilizan el método Simplex para alcanzar soluciones óptimas en los problemas planteados.

Por otro lado el Algoritmo Memético implementado, que combina la búsqueda basada en poblaciones y la mejora local con sus pautas de diseño alcanza la solución óptima en los problemas enunciados en este artículo.

A lo largo del artículo se explica cómo utilizar cada uno de los métodos, para que se conozcan cada vez más y puedan utilizarse en el entorno empresarial con el objetivo de ayudar en la toma de decisiones.

Referencias

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, p. 649-656.

Cotta C. 2007. “Una Visión General de los Algoritmos Meméticos”. 1ª Edición. *Procedimientos Metaheurísticos en Economía y Empresa*. España, ASEPUMA. p. 139-166.

Darwin C. 1859. *On the Origin of Species by Means of Natural Selection*, London, Murray. 502 p. ISBN: N/A.

Davis L. 1991 *Handbook of Genetic Algorithms*, 1st Edition. New York, Van Nostrand Reinhold, 385 p. ISBN: 978-0442001735.

Holland J. 1975. *Adaptation in Natural and Artificial Systems*, 1st Edition, USA, Ann Arbor, University of Michigan Press. 228 p. ISBN: 978-0262581110

Michalewicz Z. 1996. *Genetic algorithms + data structures = evolution programs*. 1st Edition. USA, Springer. 387 p. ISBN: 978-3540606765

Michalewicz Z, Fogel D. B. 2004. *How to Solve It: Modern Heuristics*. 2nd Edition Revised and Extended. USA, Springer. 467 p. ISBN: 978-3540224945

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.

Pearl J. 1984. *Heuristics. Intelligent Search Strategies for Computer Problem Solving*. 1st Edition. Massachusets: Addison Wesley. 399 p. ISBN: 978-0201055948

Ríos S. 1996. *Investigación Operativa. Programación lineal y aplicaciones*. 1ª Edición. Madrid, Editorial Centro de Estudios Ramón Areces S.A. ISBN: 84-8004-206-0

Syswerda G. 1989. “Uniform crossover in genetic algorithms”. *Proceedings of the 3rd International Conference on Genetic Algorithms*, USA, p. 2-9.

Shuhei K. 2003. “A Genetic Algorithm with Distance Independent Diversity Control for High Dimensional Function Optimization”. *Transactions of the Japanese Society for Artificial Intelligence*. Vol 18, p. 193-202.

Villanueva J. 2008. "La simulación de procesos, clave en la toma de decisiones", *DYNA - Ingeniería e Industria*. Vol. 83, N° 4, p. 221-227.

Whitley D. 1994. "A Genetic Algorithm Tutorial" *Statistics and Computing*. Vol. 4. Number 2, p. 65-85.

NOTA ACLARATORIA

Por solicitud de D. Jesús M^a Larrañaga Lesaka, autor del artículo “Algoritmos meméticos en problemas de Investigación Operativa”, publicado originalmente en el número 18 (diciembre de 2011) de la “Revista de Dirección y Administración de Empresas” (ISSN 1135-2051), páginas. 189 – 208, se rectifica su artículo en los siguientes puntos.

En el punto de RESUMEN (pg.190) dice textualmente:

“Los tres métodos conseguirán buenos resultados ante problemas sencillos de Investigación Operativa, sin embargo, se propone otro problema en el cual el Algoritmo Memético y la herramienta Solver de Microsoft Excel, alcanzaran la solución optima. La resolución del problema utilizando PHPSimplex resultará inviable.”

Siendo rectificado por:

“Los tres métodos conseguirán buenos resultados ante problemas de Investigación Operativa.”

En el punto 6 de “Resolución del problema de la elaboración de zumos” (págs. 202 y siguientes), concretamente al final del punto 6.1 se dice textualmente:

“Una vez llegado a este punto, tanto dando al botón de continuar o de Solución Directa que observamos en la Figura 2, el programa se bloquea y no es capaz de proporcionar ninguna solución”

Siendo rectificado por:

“Una vez llegado a este punto, dando al botón de continuar o de Solución Directa que observamos en la Figura 3, el programa nos proporciona un resultado óptimo igual al alcanzado por el Algoritmo Memético propuesto en el punto 6.2 de este artículo.”

Y en el punto 7 de Conclusiones (pág. 207), se dice textualmente:

“PHPSimplex y Solver de Microsoft Excel utilizan el método Simplex para intentar alcanzar soluciones óptimas en los problemas planteados, sin embargo, el primero de ellos se atasca en uno de los problemas propuestos.”

Siendo rectificado por:

“Se presentan en este artículo tres maneras de resolver problemas de Investigación Operativa, concretamente de Programación Lineal. PHPSimplex y Solver de Microsoft Excel utilizan el método Simplex para alcanzar soluciones óptimas en los problemas planteados.

Por otro lado el Algoritmo Memético implementado, que combina la búsqueda basada en poblaciones y la mejora local con sus pautas de diseño alcanza la solución óptima en los problemas enunciados en este artículo. “

La versión on line ha sido actualizada.

Se agradecen las aportaciones realizadas por D. Juan José Ruiz Ruiz CEO de PHPSimplex.