# IMPLEMENTACIÓN DE UN ALGORITMO DE OPTIMIZACIÓN PARA SISTEMAS DISCRETOS FUNDAMENTADO EN LA TÉCNICA DE ENJAMBRE DE PARTÍCULAS

**CARLOS EDUARDO GÓMEZ MENESES**

**OSCAR EDUARDO PÉREZ LOZANO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENIERIAS FISICO - MECANICAS**

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES**

**BUCARAMANGA**

**2012**

# IMPLEMENTACIÓN DE UN ALGORITMO DE OPTIMIZACIÓN PARA SISTEMAS DISCRETOS FUNDAMENTADO EN LA TÉCNICA DE ENJAMBRE DE PARTÍCULAS

**CARLOS EDUARDO GÓMEZ MENESES**

**OSCAR EDUARDO PÉREZ LOZANO**

**Trabajo de grado para optar al título de INGENIERO ELECTRÓNICO**

**RODRIGO CORREA, Ph.D.**

**DIRECTOR DEL TRABAJO DE GRADO**

**Ing. IVÁN MAURICIO AMAYA, Ph.D(c)**

**CODIRECTOR TRABAJO DE GRADO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENIERIAS FISICO - MECANICAS**

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES**

**BUCARAMANGA**

**2012**

## *DEDICATORIA*

*A mi mamá Nelly por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor. ¡Gracias por darme la vida! ¡Te amo demasiado!*

*A la vida por permitirme disfrutar de este maravilloso mundo y darme una familia muy especial, además de colocar en el camino personas tan importantes para mí.*

*A mis papás Carlos y Fernando a quienes les debo las más grandes enseñanzas de mi vida, le agradezco el cariño, la comprensión, la paciencia y el apoyo que me brindaron para culminar mi carrera profesional.*

*A mi hermana Leidy Milena por su cariño y comprensión incondicional en los momentos especiales que compartimos. ¡Hermanita te quiero mucho!*

*A Zully, por su amor, apoyo y compañía en cada etapa del camino recorrido juntos y también en aquellos momentos difíciles.*

*A mi amigo y compañero de tesis, Oscar por ser mi apoyo en la elaboración de este proyecto, gracias por la paciencia y los momentos compartidos.*

*A mis amigos más cercanos, que siempre me han acompañado y con los cuales he contado en grandes momentos de mi vida.*

*A mis profesores que me han acompañado durante el largo camino, brindándome siempre su orientación con profesionalismo en la adquisición de conocimientos, y cuya motivación contribuyó en gran medida a mi formación profesional culminando en la elaboración de esta tesis.*

*A la Universidad Industrial de Santander, a la Facultad de Ingenierías Físico - Mecánicas y en especial a la Escuela de Ingeniería Electrónica que me dieron la oportunidad de formar parte de ellas. ¡Gracias!*

*Carlos Eduardo Gómez Meneses*

*En primer lugar a Dios todo poderoso por permitirme salir adelante y ayudarme en todo momento a culminar con éxito esté logro en mi vida profesional.*

*A mi madre Adelaida, por su paciencia, apoyo, cariño y dedicación incondicional para la consecución de mis metas. ¡Te Amo Mamita Linda!*

*A mi padre Eduardo, por su orientación, motivación y apoyo en todos los momentos que hacen de mi vida personal y profesional más exitosa.*

*A mi abuelita Rosalba que me ayudo en cada momento a crecer como persona y ser mejor día a día. ¡Gracias por estar siempre ahí!*

*A mis hermanos Samuel Fabián y Andrea Katherine quienes con su apoyo hicieron de esté un logro común y un punto de partida en sus vidas como adultos.*

*A Juliana Rangel por su amor, paciencia, cariño esfuerzo e invaluable ayuda.*

*A mi compañero de proyecto por aguantarme, tolerarme y colaborarme en cada etapa de este trabajo.*

*A mis profesores quienes me ayudaron a crecer no solo profesionalmente sino también como persona a través de su experiencia y conocimiento.*

*A mis amigos por su colaboración en los momentos más difíciles, su apoyo y sus palabras de aliento fueron invaluables.*

*Oscar Eduardo Pérez Lozano*

## AGRADECIMIENTOS

Agradecemos especialmente a la Universidad Industrial de Santander, a la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones por haber aportado mucho en nuestra formación y permitirnos sus instalaciones para el desarrollo de nuestro proyecto de grado. A nuestro director de proyecto PhD. Carlos Rodrigo Correa y codirector de proyecto Ing. Iván Mauricio Amaya quienes con su dedicación, conocimiento, y experiencia permitieron la consecución de este importante logro en nuestras carreras profesionales y en general a todos los docentes que tuvieron algo que ver con nuestra formación académica. A nuestros familiares y amigos porque sin su continuo apoyo no hubiéramos logrado esta importante meta.

Los Autores.

## RESUMEN


**TITULO:** Implementación de un algoritmo de optimización para sistemas discretos fundamentado en la técnica de enjambre de partículas[*]

**AUTORES:** Carlos Eduardo. Gómez Meneses, Oscar Eduardo Pérez Lozano[**]

**PALABRAS CLAVE:** PSO Continuo, PSO combinatorio, función objetivo, ecuaciones diofánticas, optimización.

**CONTENIDO:**

Utilizar estrategias para enfrentar situaciones reales de ingeniería, es una tarea desarrollada en los últimos años con algoritmos metaheurísticos. Un ejemplo de estas metaheurísticas es el algoritmo PSO (Particle Swarm Optimization), que realiza una búsqueda de un óptimo local, y uno global con cada iteración, encontrando valores de velocidad y posición en cada partícula e informándole a si está cerca de la solución del problema. En caso de no estarlo, se remite al espacio de búsqueda para seguir al valor más cercano a la respuesta.

En este documento, se describe la implementación de un algoritmo evolutivo (PSO combinatorio), para resolver algunos problemas de matemática e ingeniería relacionados con las ecuaciones diofánticas. Entre las características más importantes, se resalta la eficiencia del algoritmo implementado, para encontrar la solución.

En ingeniería electrónica, se mencionan algunas aplicaciones de las ecuaciones diofánticas, y se motiva a utilizar el algoritmo como alternativa de solución. La evaluación de la eficiencia, se realiza ejecutando el algoritmo para solucionar problemas de matemática discreta con respuesta conocida, haciéndose al final un análisis de resultados. De forma ilustrativa, se presentan dos problemas de la teoría de control en el diseño de controladores, que permite evidenciar la aplicación, en principio, a cualquier sistema de ecuaciones diofántico.


---

[*] Proyecto de Grado

[**] Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Rodrigo Correa Ph.D. Codirector: Ing. Iván Amaya Ph.D(c).

## ABSTRACT

**TITLE:** Implementation of an optimization algorithm for discrete systems based on particle swan technique[*]

**AUTHORS:** Carlos Eduardo. Gómez Meneses, Oscar Eduardo Pérez Lozano[**]

**KEYWORDS:** PSO, combinatorial PSO, fitness function, diophantine equations, optimization.

**CONTENT:**

Using strategies for facing engineering real situations has been a task developed in recent years with metaheuristic algorithms. An example of these metaheuristics is the Particle Swarm Optimization (PSO) algorithm, which performs a search for a local, and a global, optimum on each iteration. Thus, it finds position and speed values for each particle, and it lets them know whether they are close to the solution, or not. In the latter case, it uses the search space to keep looking for a closer value.

In this document we describe the implementation of an evolutionary algorithm (combinatory PSO), to solve some mathematical and engineering problems related to Diophantine equations. The most important features are the efficiency of the algorithm implemented for finding the solution.

For the Diophantine equations we also mention some applications in electronic engineering and motivate the use of the algorithm as an alternative of solution. The evaluation of the efficiency is performed by running the algorithm for solving discrete mathematical formulations with known results, making as final comparative analysis an illustration for two problems of control theory submitted in the design of controllers, which allow to test the application, in principle, for any system of Diophantine equations.


---

[*] Thesis

[**] Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Rodrigo Correa Ph.D. Codirector: Ing. Iván Amaya Ph.D(c).

**CONTENIDO**

## LISTA DE TABLAS

## LISTA DE FIGURAS

## LISTA DE ANEXOS

# INTRODUCCION

En ingeniería, aparecen algunos problemas de optimización que se solucionan a través de métodos metaheurísticos, el proceso de solución sigue un modelo evolutivo inspirado en la naturaleza, que se desarrolla por medio de ciclos iterativos, solucionando un problema guiado por una heurística subordinada, que combina los conceptos para explorar y explotar adecuadamente el espacio de búsqueda. Un ejemplo de estas metaheurísticas es el algoritmo PSO (Particle Swarm Optimization), que realiza una búsqueda de un óptimo local, y uno global, con cada iteración, informándole a cada partícula si está cerca de la solución del problema. En caso de no estarlo, se remite al espacio de búsqueda para seguir al valor más cercano a la respuesta. Se pueden resolver algunos problemas discretos al implementar una variante de este algoritmo, conocida como PSO combinatorio. Este tipo de optimización maneja sus variables dentro de un conjunto finito de posibilidades, y busca un objeto en el menor número de configuraciones posibles.

En ingeniería electrónica, se mencionan algunas aplicaciones de las ecuaciones diofánticas, y se motiva a utilizar el algoritmo como alternativa de solución. La evaluación de la eficiencia, se realiza ejecutando el algoritmo para solucionar problemas de matemática discreta con respuesta conocida, haciéndose al final un análisis de resultados. De forma ilustrativa, se desarrollan dos problemas de la teoría de control en el diseño de controladores, que permite evidenciar la aplicación, en principio, a cualquier sistema de ecuaciones diofántico.

Este informe recopila los principales resultados logrados. El capitulo dos contiene la descripción del trabajo de grado, incluidos el objetivo general y los objetivos específicos planteados desde el inicio. El capitulo tres desarrolla los principales

conceptos teóricos en que se fundamenta este trabajo de grado, en modalidad de investigación. La implementación del algoritmo combinatorio se explica en el capítulo cuatro. Las pruebas del algoritmo implementado se hacen resolviendo problemas típicos, de diversas áreas y contrastando sus resultados frente a la solución analítica de los mismos como aparece en el capítulo cinco. Se resolvieron dos ejercicios de ingeniería electrónica, más específicamente en el área de control de procesos y a título de ejemplo, todo ello se presenta en el capítulo seis. Los capítulos siete y ocho resumen las principales conclusiones y recomendaciones.

# 2. DESCRIPCION DEL TRABAJO DE GRADO

Este informe de investigación se realiza con el fin de recopilar el proceso seguido y los principales resultados alcanzados en desarrollo del trabajo de grado: “Implementación de un algoritmo de optimización para sistemas discretos fundamentado en la técnica de enjambre de partículas”. Así mismo, se evidencia que se cumplieron todos los objetivos presentados en el plan de trabajo de grado, enunciados en las siguientes subsecciones.

## 2.1 OBJETIVO GENERAL

Contribuir a una eventual creación en el grupo CEMOS de una línea relacionada con la optimización de sistemas discretos a través de la implementación de un algoritmo combinatorio en Matlab con enjambre de partículas.

## 2.2 OBJETIVOS ESPECÍFICOS

Para el cumplimiento del objetivo general del proyecto se requiere lo siguiente:

- Implementar una estrategia de optimización combinatoria con enjambre de partículas, CPSO, para resolver problemas de ingeniería electrónica donde se presenten ecuaciones diofánticas.

- Desarrollar ejercicios de carácter ilustrativo que permitan verificar las características de este algoritmo en la solución de problemas discretos.

- Comparar los resultados obtenidos frente a los encontrados con métodos tradicionalmente utilizados para resolver ecuaciones diofánticas.

# 3. FUNDAMENTOS TEÓRICOS

En esta sección se muestran los conceptos relacionados con el algoritmo de optimización por enjambre de partículas (PSO), así como, la función objetivo, la modificación del algoritmo para resolver problemas combinatorios (CPSO) con ecuaciones diofánticas.

## 3.1 PSO (PARTICLE SWARM OPTIMIZATION) CONTINUO

PSO fue propuesto por Kennedy y Eberhart en 1995, y se encuentra inspirado en el comportamiento colectivo animal. Básicamente, busca desplazar a las partículas por el espacio de búsqueda, que al ser evaluado en la función objetivo $F(x)$ permite obtener una medida del nivel de ajuste a una solución óptima candidata, con lo que se puede optimizar el sistema al modificar la velocidad y posición de las mismas. Se asume que el espacio de búsqueda de n dimensiones se define por dos vectores para cada partícula:

$$X_i = (x_{i1}, x_{i2}, \ldots x_{1n})$$

$$V_i = (v_{i1}, v_{i2}, \ldots v_{1n})$$

donde $X_i$ corresponde a la posición de la partícula $i$, en cada una de las $n$ dimensiones, y $V_i$ a su velocidad. El algoritmo inicia con una distribución de partículas aleatoria sobre el espacio de búsqueda. Luego de esto, se calcula su ajuste (evaluando en $F(x)$) y se actualiza la posición y velocidad de cada una ($X_{i+1}$

y $V_{i+1}$, respectivamente), a través de las ecuaciones (3.1) y (3.2), donde $loc$ representa el óptimo local de cada partícula y $glo$ el del enjambre. Adicionalmente, $c1, c2$ son los coeficientes de confianza local y global, respectivamente, $r1, r2$ son números aleatorios, uniformemente distribuidos entre 0 y 1, y $w$ es el factor de inercia que evita que las partículas se dispersen demasiado [6], [17], [24], [26].

$$V_{i+1} = V_i * w + c1 * r1 * (loc - X_i) + c2 * r2 * (glo - X_i) \quad (3.1)$$

$$X_{i+1} = X_i + V_i \quad (3.2)$$

## 3.2 FUNCIÓN OBJETIVO

Como su nombre lo indica, es la función que representa el sistema objeto de la optimización. Para este caso, se toma como la sumatoria de cada una de las ecuaciones que modelan el problema al cuadrado, de la forma mostrada en (3.3) [24].

$$F(x) = \sum(Ecuacion\ 1)^2 + (Ecuacion\ 2)^2 + \cdots + (Ecuacion\ N)^2 \quad (3.3)$$

A manera de ejemplo:

$$a1 * x + b1 * y = m \quad \rightarrow \quad a1 * x + b1 * y - m = 0 \quad (3.4)$$

$$a2 * x + b2 * y = n \quad \rightarrow \quad a2 * x + b2 * y - n = 0 \quad (3.5)$$

$$F(x) = \sum((a1 * x + b1 * y - m)^2 + (a2 * x + b2 * y - n)^2) \quad (3.6)$$

## 3.3 PSO COMBINATORIO

Es una variante de PSO utilizada para resolver problemas con ecuaciones de variables discretas, donde los números enteros son las únicas posibles soluciones. Al igual que en PSO, existen dos vectores $X_i$ y $V_i$ correspondientes a la posición y velocidad de cada partícula, en cada iteración. El primero, inicialmente, es un vector de números aleatorios enteros en el intervalo válido de la solución, y el vector inicial de velocidad son números aleatorios enteros asociados a cada partícula. Cuando los problemas son multivariables aparecen matrices de velocidad y posición, puesto que la solución es un conjunto de valores asociados a cada incógnita de la ecuación [10].

CPSO se diferencia de PSO en que la nueva velocidad y posición dependen tanto de una ecuación, como de una toma de decisión, que escoge entre el local y global para la nueva iteración. Asuma que existe un vector $y_i = (y_{i1}, y_{i2}, \ldots y_{1n})$ que permite la transición del PSO convencional al combinatorio y que toma los valores de -1,1 o 0, dependiendo si en la iteración actual el valor del vector de posición es igual o diferente al local y al global, como muestra la ecuación (3.7). Posteriormente, se actualiza la velocidad con la ecuación (3.8) [10].

$$y_i = \begin{cases} 1 & \leftrightarrow \quad X_i = glo \\ -1 & \leftrightarrow \quad X_i = loc \\ 0 & \leftrightarrow X_i \neq glo \neq loc \\ -1 \; o \; 1 & \leftrightarrow X_i = glo = loc \end{cases} \quad (3.7)$$

$$V_{i+1} = V_i * w + c_1 * r_1 * (-1 - y_i) + c_2 * r_2 * (1 - y_i) \quad (3.8)$$

Así mismo, se debe calcular el parámetro de decisión "*Dummy*", o vector $B_i = (B_{i1}, B_{i2}, \ldots, B_{im})$, con la ecuación (3.9). Este parámetro decide si la posición de la partícula siguiente es igual al global, al local o a un número aleatorio entero que esté dentro del rango del espacio de búsqueda, según lo mostrado en la ecuación (3.10). $\alpha$ es una constante del problema que define la intensificación

(nueva posición igual al óptimo global o local) y la diversificación (nueva posición igual a un número aleatorio) [10], [23].

$$B_i = y_i + V_{i+1} \quad (3.9)$$

$$X_{i+1} = \begin{Bmatrix} glo & \leftrightarrow & B_i > \alpha \\ loc & \leftrightarrow & B_i < -\alpha \\ \#\ aleatorio\ entero & \leftrightarrow & -\alpha \leq B_i \leq \alpha \end{Bmatrix} \quad (3.10)$$

## 3.4 ECUACIONES DIOFÁNTICAS

Dentro de la variedad y potenciales aplicaciones en sistemas discretos, se decidió abordar el problema de las ecuaciones diofánticas que se trata seguidamente. Conceptualmente son aquellas, generalmente multivariables, con coeficientes enteros que admiten únicamente soluciones enteras. Su nombre es en reconocimiento al matemático griego del siglo III dC. Diofanto de Alejandría, quien fue uno de los pioneros en escribir notación simbólica en matemáticas. Se define una ecuación diofántica lineal de $n$ incógnitas de acuerdo a la ecuación (3.11), denotando $a_1, a_2, \dots, a_n$ números enteros conocidos y $x_1, x_2, \dots, x_n$ números enteros a encontrar [7].

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \quad (3.11)$$

Para el caso de una ecuación con dos incógnitas, las posibles soluciones de la ecuación son:

$$a * x + b * y = c \quad (3.12)$$

donde $a, b, c$ son números enteros conocidos. Es de gran importancia tener en cuenta que esta ecuación sólo tiene solución sí el máximo común divisor de $a$ y $b$ es un divisor de $c$, con lo que se puede escribir una solución general para

esta ecuación de la forma (3.13), donde $\beta$ es un número entero cualquiera. Se enuncia que $x_0$, $y_0$ son dos soluciones de la ecuación.

$$x = x_0 + \beta * \frac{b}{d} \qquad y = y_0 - \beta * \frac{a}{d} \quad (3.13)$$

Existen dificultades para resolver este tipo de ecuaciones con múltiples variables, ya sean ecuaciones pitagóricas de la forma $x^2 + y^2 = z^2$, ecuaciones polinomiales $P(x) - Q(x) = 0$ ó sistemas de ecuaciones lineales, llamando la atención de varios matemáticos, a través de los años, quienes crearon métodos o formas para encontrarles solución. Entre los más conocidos se encuentra el algoritmo de Euclides (algoritmo de división polinómica), el método basado en las transformaciones elementales de las matrices y el método de coeficientes indefinidos. Por esta razón es importante encontrar un método capaz de resolver sistemas de ecuaciones diofánticas y enfrentar así problemas de la teoría del algebra de control discreto por medio de un algoritmo, que utilice la matemática combinatoria ofreciendo resultados eficientes. Matemáticos como Demjanenko, Ramanujan, Euler, Hardy, y Carmichael han dedicado parte de sus trabajos, a la creación de métodos algebraicos para la solución particular de algunas ecuaciones diofánticas de orden superior, basados en los principios planteados por Diofanto [7], [8], [25].

En la actualidad, este tipo de ecuaciones han tomado protagonismo en muchos campos de ingeniería. En el diseño de mecanismos y engranajes cilíndricos, en la mecánica industrial, el problema se dirige a la relación entre el número de dientes de dos piñones y únicamente se puede abordar con las soluciones enteras positivas de un sistema de ecuaciones diofántino. Así mismo, se encuentran frecuentemente en problemas de minimización o maximización de costos donde resultados en el campo de los reales no son una opción, ya que su objeto son los números enteros. En problemas de ingeniería electrónica, la solución de

problemas de control es necesaria, puesto que a través de ellas se puede diseñar un controlador que se acople a las necesidades del modelo [2].

## 4. IMPLEMENTACIÓN DEL ALGORITMO

Este capítulo del informe se desarrolla con fin de cumplir el siguiente objetivo específico:

- Implementar una estrategia de optimización combinatoria con enjambre de partículas CPSO para resolver problemas de ingeniería electrónica donde se presenten ecuaciones diofánticas.

La implementación del algoritmo de optimización para sistemas discretos, se fundamenta en la técnica de enjambre de partículas y es el elemento más importante del presente informe. La primera versión del algoritmo computarizado para el modelado y solución de este tipo de problemas, se hizo en Matlab (se recomienda trabajar en una versión 2008 o superior para la ejecución del mismo). En la tabla 1 se definen todas las variables y constantes utilizadas en este capítulo.

| | |
|---|---|
| **np** | Número de partículas |
| **ni** | Número máximo de iteraciones |
| **nd** | Número de variables |
| **X** | Posición de las partículas |
| **V** | Velocidad de las partículas |
| **fx** | Función objetivo |
| **w** | Ponderación de inercia |
| **c1** | Coeficiente de confianza local |
| **c2** | Coeficiente de confianza global |
| **var1** | Mínimo valor de fx en la partícula |

| | |
|---|---|
| **var2** | Posición de var1 |
| **loc** | Óptimo local de la partícula |
| **floc** | Función objetivo evaluada en los óptimos locales de la iteración |
| **glo** | Óptimo global del enjambre |
| **fglo** | Función objetivo evaluada en el optimo global del enjambre o fx en var2 |
| **final** | Valor de parada del algoritmo, indica cuando el 99% tienen la solución |
| **r1** | Número aleatorio de 0 a 1 |
| **r2** | Número aleatorio de 0 a 1 |
| **alfa** | Parámetro de decisión |
| **duración** | Tiempo de cómputo del algoritmo |
| **Cont** | Es el contador de las soluciones para los problemas que tienen múltiples |
| **Comp** | Permite comparar las soluciones arrojadas por el algoritmo |

**Tabla 1. Constantes y variables utilizadas en el capítulo 4**

El algoritmo se diseña, para que el usuario defina el número de partículas, el número máximo de iteraciones y la cantidad de incógnitas del problema; en este ejemplo son **x, y, z,** es decir **3**. Se escoge la posición y velocidad como dos matrices (tamaño **np** x **nd**) de números aleatorios enteros. Se introduce la función objetivo teniendo en cuenta siempre, que **x1** es la primera variable, en este caso **x=x1**, **xn** es la última variable en este caso **z = x3**. En la primera iteración, el óptimo local es igual a **x**, y la función del óptimo local es igual a **fx**. Se definen constantes de PSO combinatorio e inicializan variables utilizadas más adelante. Se encuentran los valores iníciales de **var1** y **var2**, para reemplazar posteriormente el optimo global del enjambre, y el **fglo** respectivo. Por último, se pregunta si **final** es menor o igual a **np**, si esto es correcto, se continúa al globo **B**, si es incorrecto, se imprime la solución encontrada, el tiempo de cómputo y el ciclo de trabajo. Este proceso se representa en la figura 1.

En este párrafo, se explica la segunda parte del diagrama. El globo **B** orienta el punto de conexión entre las figuras 1 y 2. Posteriormente se definen **r1**, **r2**, e

inicializan dos ciclos: “for”, se acumula **i** hasta **np** y **j** hasta **nd** respectivamente. Se pregunta si **x** es igual al óptimo local y global simultáneamente, si es correcto, se iguala **y** a **1** o **-1** aleatoriamente, si no es correcto se pregunta ahora si **x** es igual al global, si es cierto, se iguala **y** a **1**, en caso contrario se pregunta si **x** es igual al local, si es correcto se iguala **y** a **-1,** si es incorrecto se iguala **y** a **0**. Cuando se termine de acumular **np** se debe dirigir al globo **C** para conectar a la figura 3. Este proceso es representado en la figura 2.

El globo **C** es el punto de conexión entre las figuras 2 y 3. La figura 3 corresponde a la tercera parte del diagrama de flujo, e inicia acumulando **i** y **j** hasta **np** y **nd** respectivamente. Después se procede a hallar la nueva velocidad y el parámetro **B** “dummy” con las fórmulas de (5), (6), (7), (8), (9), (10) de [10]. Se acumula de nuevo **i** y **j** hasta **np**, **nd** respectivamente. Se pregunta si **B** es mayor que **alfa**, en caso correcto, **x** en la nueva iteración toma el valor del óptimo global en la iteración actual, en caso incorrecto, se pregunta si **B** es menor que **alfa**, en caso correcto, **x** en la nueva iteración toma el valor del óptimo local en la iteración actual, en caso incorrecto, se asigna **x** como un número aleatorio entero que esté dentro del espacio de búsqueda. Cuando se termina de acumular el segundo ciclo “for” en **i,** se sigue al globo **D**.

El globo **D** es el punto de conexión entre las figuras 3 y 4. Las figuras 4 y 5 corresponden a la cuarta y quinta parte del diagrama de flujo respectivamente, e inician acumulando **i** hasta **np.** Después se procede a encontrar el **fx** en la nueva iteración, para ello, se debe introducir la función objetivo en la iteración siguiente, es decir k+1; para esto se tiene en cuenta, que **i** contiene la información del número de partículas. Por tal motivo, la primera variable, en este caso **x** es un vector de la forma **x(i,1,k+1)**. La última variable es un vector de la forma **x(i,n,k+1)** con **n** variables, en este ejemplo la variable es **z** y es de la forma **x(i,3,k+1)**. Se

acumula **i** y **j** hasta **np** y **nd** respectivamente, se pregunta si la función objetivo evaluada en **x** de la nueva iteración, es igual a la función objetivo evaluada en el óptimo local de la iteración actual **floc**; en caso correcto, se asume el óptimo local de la nueva iteración como el **x** de la nueva iteración; en caso incorrecto, se iguala el nuevo óptimo local al óptimo local actual (mínimo). Se acumula **i** hasta **np.** Se escribe la función **floc**, que corresponde a los óptimos locales evaluados en el **fx** de la nueva iteración, es decir en **k+1**, se tiene en cuenta que **i** contiene la información del número de partículas. La primera variable en este caso **x**, es un vector de la forma **loc(i,1,k+1)**, la última variable es un vector de la forma **loc(i,n,k+1)**, con **n** variables, en este caso **z** y es de la forma **loc(i,3,k+1)**. Se encuentra **var1** y **var2**.

Se define el óptimo global como el óptimo local en la posición **var2**. Se iguala **fglo** a **var1**. Se pregunta si **final=1** y **final2=0**, en caso correcto, se dice que **final2=k**, en caso incorrecto, se pregunta si **k=ni**, en caso correcto se dice que **final=np**, en caso incorrecto, se pregunta si **cont=1**, en caso correcto se iguala **final** a **np,** en caso incorrecto se dirige al globo **A**. El globo entrega la secuencia a la figura 1.

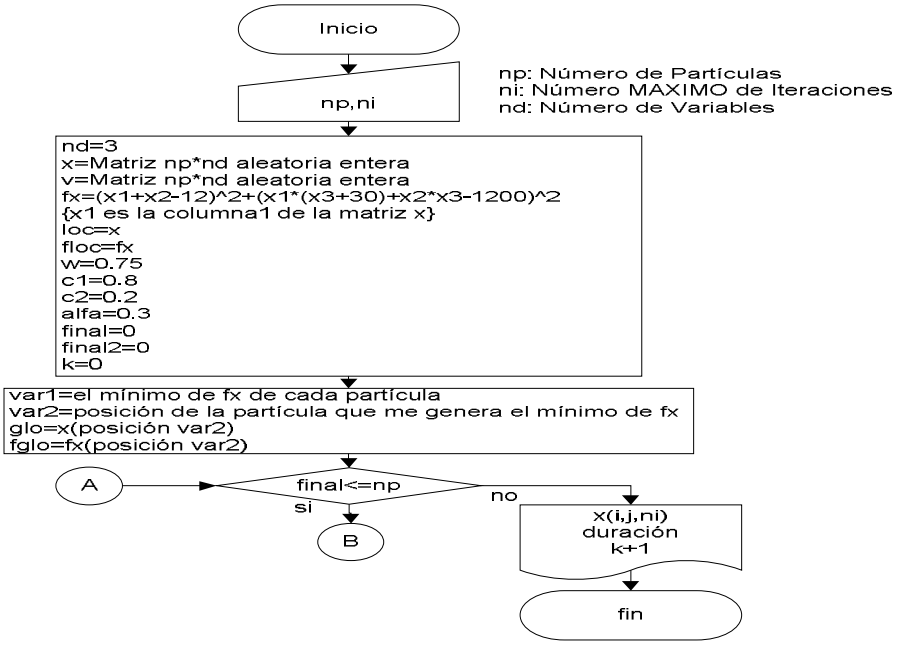


**Figura 1. Primera parte del diagrama de flujo del algoritmo**

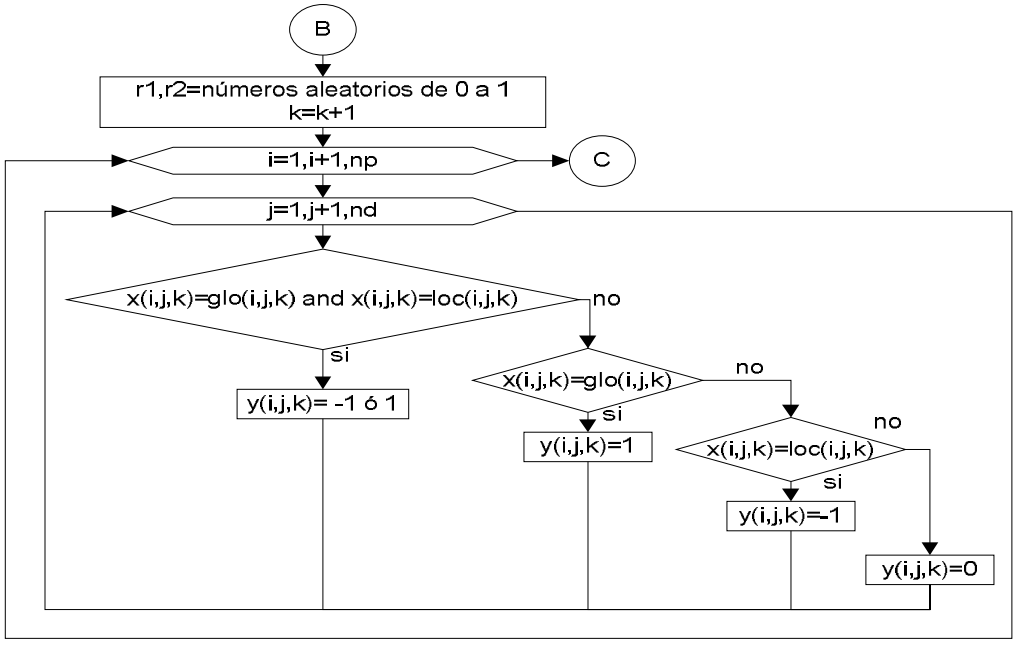

**Figura 2. Segunda parte del diagrama de flujo del algoritmo**

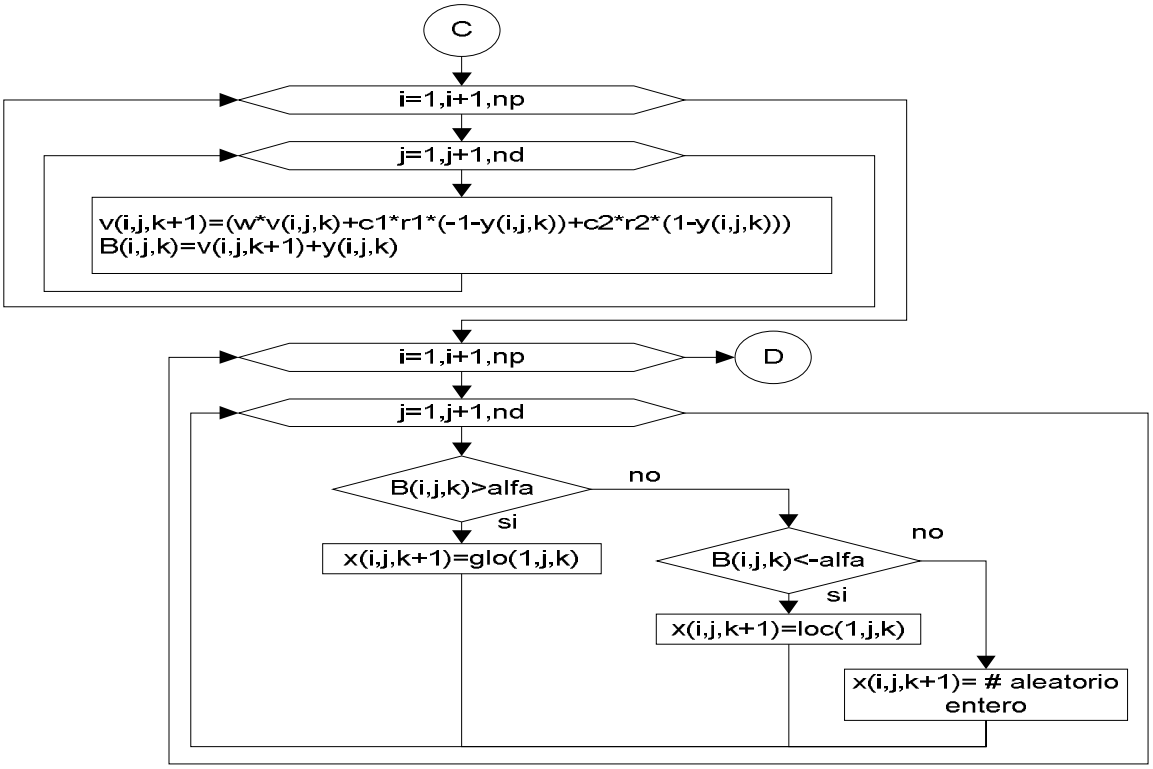


**Figura 3. Tercera parte del diagrama de flujo del algoritmo**

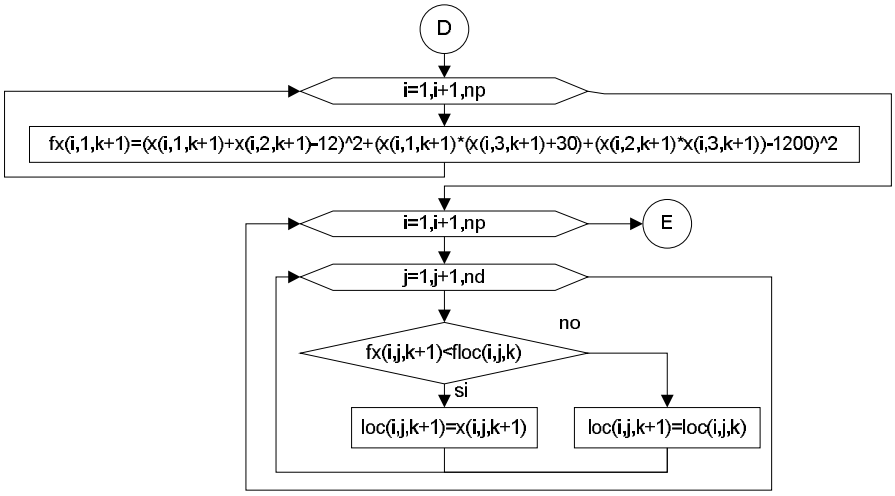


**Figura 4 Cuarta parte del diagrama de flujo del algoritmo**

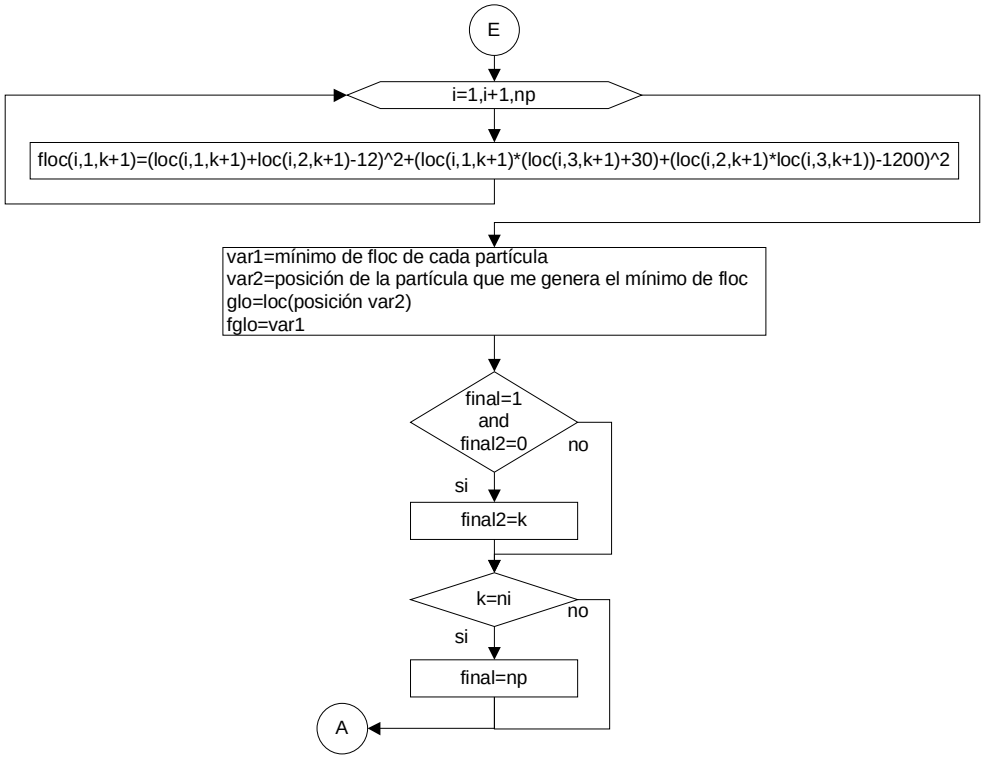


**Figura 5 Quinta parte del diagrama de flujo del algoritmo**

## 5. EVALUACIÓN DEL ALGORITMO A TRAVÉS DE EJERCICIOS ILUSTRATIVOS

Esta sección se desarrolla con fin de cumplir los objetivos del trabajo de grado:

- Desarrollar ejercicios de carácter ilustrativo que permitan verificar las características de este algoritmo en la solución de problemas discretos.
- Comparar los resultados obtenidos frente a los encontrados con métodos tradicionalmente utilizados para resolver ecuaciones diofánticas.

Debido a la gran cantidad de aplicaciones de las ecuaciones diofánticas, tanto en la matemática discreta como en problemas de ingeniería, en la evaluación del algoritmo se ejecutaron diez ejemplos de ecuaciones con enteros, en un equipo de cómputo con procesador AMD Turion X2 Dual Core RM-72 de 2.1Ghz, memoria RAM de 4Gb. El ciclo de iteración dio un valor de 0,022 segundos/ iteración. Para todos los ejemplos de esta sección se utilizó: w = 0.75 [15], c1 = 0.8, c2 = 0.2, alfa = 0.3 [10].

## 5.1 EJEMPLO: NÚMEROS CONSECUTIVOS

***Enunciado del problema:*** Pruébese que $3,4,5$ es la única solución de $x^2 + y^2 = z^2$ en enteros positivos consecutivos [11].

***Solución analítica:***

Si $x,y,z$ son tres enteros positivos consecutivos entonces se pueden escribir como $x = n-1$, $y = n$, $z = n+1$ con $n > 1$; en la ecuación (5.1) se observa el modelo del problema, la ecuación (5.2) es el desarrollo algebraico [11], en las ecuaciones (5.3) y (5.4) se encuentran las respuestas del ejercicio. La tabla 2 permite organizar estos valores.

$$(n-1)^2 + n^2 = (n+1)^2 \qquad (5.1)$$

$$n^2 - 2*n + 1 + n^2 = n^2 + 1 + 2*n \qquad (5.2)$$

$$n*(n-4) = 0 \quad \rightarrow \quad n = 4 \qquad (5.3)$$

$$x = n-1 = 3, \quad y = n = 4, \quad z = n+1 = 5 \qquad (5.4)$$

| Solución | | |
|---|---|---|
| x | y | z |
| 3 | 4 | 5 |

**Tabla 2. Solución analítica al ejemplo 5.1**

Por lo tanto los enteros consecutivos que cumplen esta condición son 3, 4, 5.

***Solución con el algoritmo:***

| Particulas | x | y | z | Duración (s) | # Iteración | Cociente |
|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 5 | 205,590 | 22224 | 0,0092508 |
| 10 | 3 | 4 | 5 | 560,981 | 76647 | 0,0073190 |
| 30 | 3 | 4 | 5 | 404,719 | 50436 | 0,0080244 |
| 50 | 3 | 4 | 5 | 1,539 | 138 | 0,0111522 |
| 70 | 3 | 4 | 5 | 62,547 | 8649 | 0,0072317 |
| 100 | 3 | 4 | 5 | 2,929 | 429 | 0,0068275 |
| 300 | 3 | 4 | 5 | 89,269 | 4941 | 0,0180670 |
| 500 | 3 | 4 | 5 | 196,746 | 6635 | 0,0296528 |
| 700 | 3 | 4 | 5 | 378,694 | 8243 | 0,0459413 |
| 1000 | 3 | 4 | 5 | 9,687 | 164 | 0,0590671 |

**Tabla 3. Resultados en algoritmo determinando cantidad de partículas**

La tabla 3, permite observar que el menor cociente entre tiempo de cómputo y número de ciclos, se presenta cuando el número de partículas es igual a 100; se realiza este procedimiento para definir una cantidad de partículas estándar en los próximos ejemplos.

| Ej. | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|
| 1 | 3 | 4 | 5 | 133,793 | 29436 |
| 2 | 3 | 4 | 5 | 0,960 | 193 |
| 3 | 3 | 4 | 5 | 136,650 | 30062 |
| 4 | 3 | 4 | 5 | 22,090 | 4913 |
| 5 | 3 | 4 | 5 | 0,900 | 193 |
| 6 | 3 | 4 | 5 | 3,810 | 855 |
| 7 | 3 | 4 | 5 | 2,370 | 521 |
| 8 | 3 | 4 | 5 | 32,490 | 7093 |
| 9 | 3 | 4 | 5 | 101,260 | 22192 |
| 10 | 3 | 4 | 5 | 313,540 | 68402 |
| 11 | 3 | 4 | 5 | 85,930 | 18894 |
| 12 | 3 | 4 | 5 | 0,891 | 171 |
| 13 | 3 | 4 | 5 | 91,300 | 16885 |
| 14 | 3 | 4 | 5 | 172,150 | 32212 |
| 15 | 3 | 4 | 5 | 391,260 | 72960 |
| 16 | 3 | 4 | 5 | 65,769 | 12176 |
| 17 | 3 | 4 | 5 | 76,076 | 449,998 |
| 18 | 3 | 4 | 5 | 1,010 | 174 |
| 19 | 3 | 4 | 5 | 117,890 | 22059 |
| 20 | 3 | 4 | 5 | 502,748 | 86061 |
| 21 | 3 | 4 | 5 | 1,392 | 235 |
| 22 | 3 | 4 | 5 | 3,516 | 625 |
| 23 | 3 | 4 | 5 | 1,138 | 206 |
| 24 | 3 | 4 | 5 | 103,151 | 19361 |
| 25 | 3 | 4 | 5 | 352,934 | 60454 |
| 26 | 3 | 4 | 5 | 71,620 | 10649 |
| 27 | 3 | 4 | 5 | 1,559 | 241 |
| 28 | 3 | 4 | 5 | 2,239 | 357 |
| 29 | 3 | 4 | 5 | 148,474 | 24680 |
| 30 | 3 | 4 | 5 | 350,205 | 58192 |
| 31 | 3 | 4 | 5 | 106,428 | 18083 |
| 32 | 3 | 4 | 5 | 49,945 | 8571 |
| 33 | 3 | 4 | 5 | 1,582 | 260 |
| 34 | 3 | 4 | 5 | 67,434 | 11530 |
| 35 | 3 | 4 | 5 | 163,234 | 27849 |
| 36 | 3 | 4 | 5 | 1,264 | 204 |

| | | | | | |
|---|---|---|---|---|---|
| 37 | 3 | 4 | 5 | 41,191 | 7023 |
| 38 | 3 | 4 | 5 | 59,904 | 10694 |
| 39 | 3 | 4 | 5 | 61,841 | 10351 |
| 40 | 3 | 4 | 5 | 47,000 | 5051 |
| 41 | 3 | 4 | 5 | 2,304 | 388 |
| 42 | 3 | 4 | 5 | 100,613 | 17026 |
| 43 | 3 | 4 | 5 | 0,801 | 123 |
| 44 | 3 | 4 | 5 | 246,735 | 42040 |
| 45 | 3 | 4 | 5 | 58,405 | 9758 |
| 46 | 3 | 4 | 5 | 80,920 | 13663 |
| 47 | 3 | 4 | 5 | 1,370 | 243 |
| 48 | 3 | 4 | 5 | 227,890 | 28610 |
| 49 | 3 | 4 | 5 | 215,417 | 36673 |
| 50 | 3 | 4 | 5 | 57,671 | 9872 |
| **Promedio** | | | | **97,621** | **17469** |

**Tabla 4. Resultados en algoritmo del ejemplo 5.1 para 100 partículas, 50 ciclos**

De la tabla 4 se concluye que el resultado es $x = 3, \quad y = 4, \quad z = 5$. Así mismo, están otros parámetros como la duración y número de iteraciones, que corresponden al tiempo de cómputo tardado por el algoritmo en encontrar la solución y a los ciclos que tomó hacerlo. Su efectividad, en términos de la respuesta encontrada, no puede ser cuestionada, puesto que, el algoritmo termina su ejecución y genera los resultados cuando todas las partículas son igual a la solución.

En la figura 6 se realiza la distribución normal de las 50 iteraciones del ejemplo, se obtiene en promedio aproximadamente 17469. En la figura 7 se observa que el punto verde (partículas) está en el cuadro rojo (solución). Para próximos ejemplos se decidió realizar 20 ejecuciones.

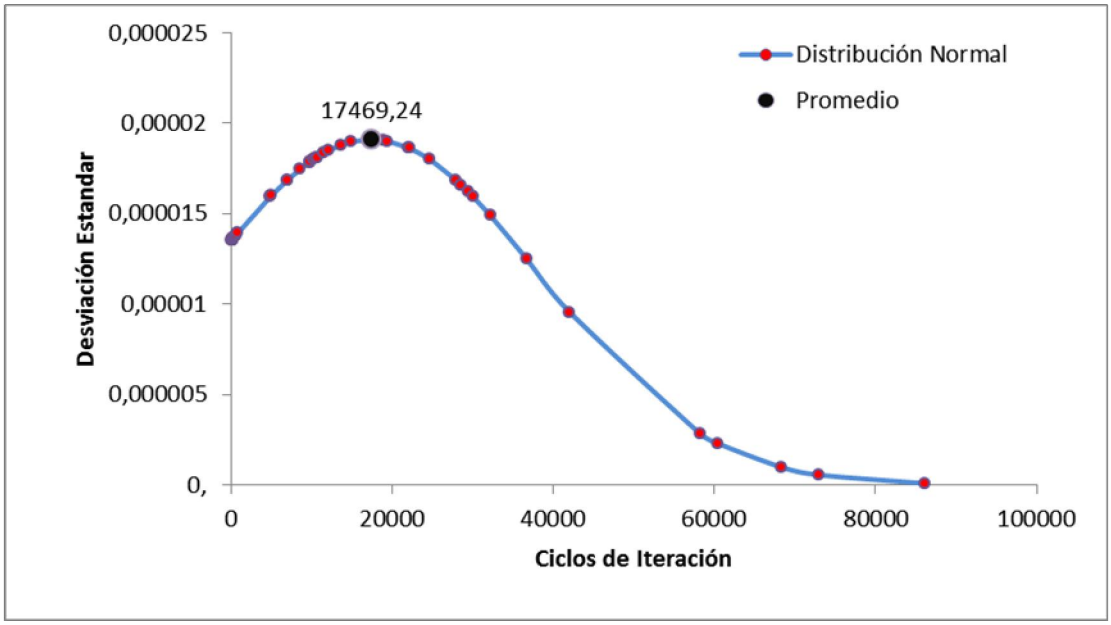


**Figura 6. Distribución normal de las iteraciones de la tabla 4**

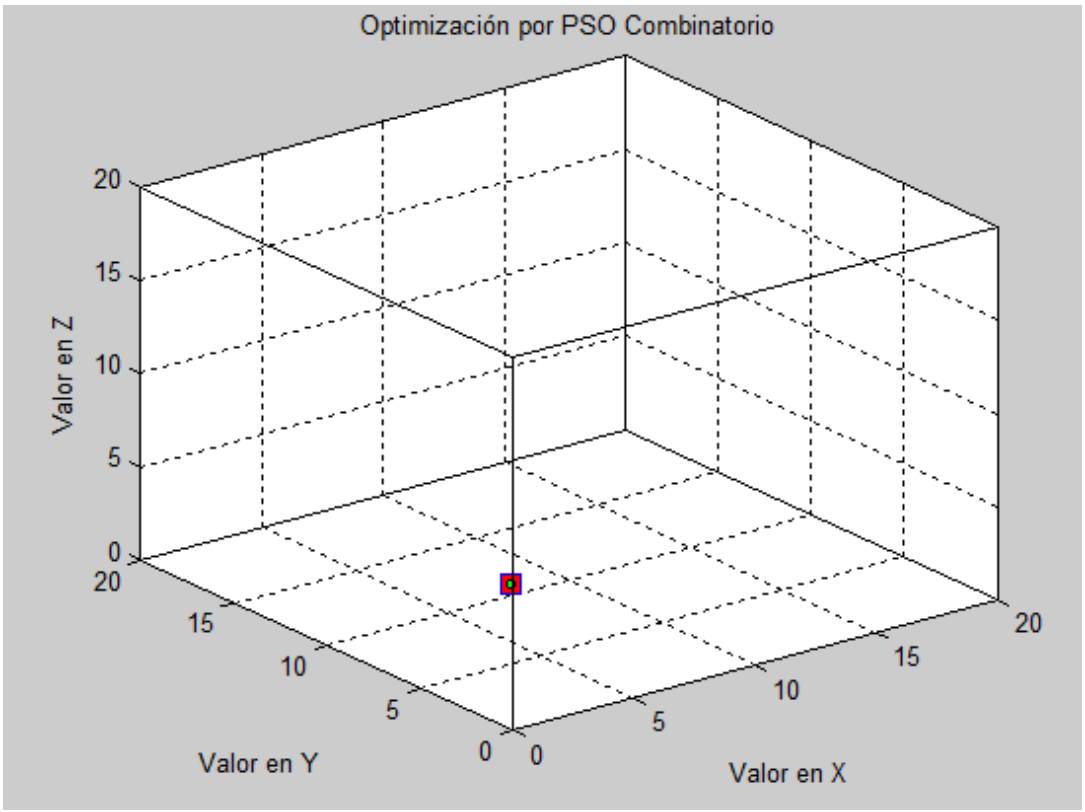


**Figura 7. Simulación en el algoritmo del ejemplo 5.1**

## 5.2 EJEMPLO: TRIPLA PITAGÓRICA

***Enunciado del problema:*** Encuéntrese todos los triángulos pitagóricos cuya área sea numéricamente igual a su perímetro [11].

***Solución Analítica:***

Sean $a, b, c$ los lados del triángulo, siendo $c$ la hipotenusa, se define la ecuación (5.5) que modela los triángulos rectángulos; si el área debe ser igual al perímetro, se plantea la ecuación (5.6). Se procede a despejar las ecuaciones (5.5), (5.6) y se desarrolla el ejercicio como ilustran las ecuaciones (5.7), (5.8), (5.9), (5.10).

$$a^2 + b^2 = c^2 \qquad (5.5)$$

$$a + b + c = \frac{a*b}{2} \qquad (5.6)$$

$$c = \frac{a*b}{2} - a - b \qquad (5.7)$$

$$a^2 + b^2 = \left(\frac{a*b}{2} - a - b\right)^2 \qquad (5.8)$$

$$a^2 + b^2 = \frac{a^2*b^2}{4} + (a+b)^2 - a*b*(a+b) \qquad (5.9)$$

$$0 = a*b*\left(\frac{a*b}{4} - 2*a*b\right) \qquad (5.10)$$

Si $a, b$ son diferentes de cero, entonces $a*b+8-4a-4b=0$, por lo tanto, se desarrollan las ecuaciones (5.11) y (5.12), así mismo de la ecuación (5.13) se debe garantizar que $(a-4) \neq 0$; en esta ecuación se observa que únicamente algunos valores de $a$ mayores que 4, garantizan que la solución esté en el rango de los enteros positivos **Z**+; es decir que $a = 5,6,8$ ó $12$.

$$b*(a-4) = 4a - 8 = 4*(a-4) + 8 \qquad (5.11)$$

$$(b-4)(a-4) = 8 \qquad (5.12)$$

$$b = 4 + \frac{8}{a-4} \qquad (5.13)$$

En el ejercicio anterior se planteó una notación para el algoritmo y se encuentran múltiples soluciones, por tanto, se hace necesario redefinir la notación para la tabulación de datos y es: $a$ = x1, x2, x3, x4; $b$ = y1, y2, y3, y4; $c$ = z1, z2, z3, z4. Se reemplazan diferentes valores de $x$ y se encuentran las ternas solución de la tabla 5.

| **Solución 1** | | |
|---|---|---|
| x1 | y1 | z1 |
| 6 | 8 | 10 |
| | | |
| **Solución 2** | | |
| x2 | y2 | z2 |
| 8 | 6 | 10 |
| | | |
| **Solución 3** | | |
| x3 | y3 | z3 |
| 12 | 5 | 13 |
| | | |
| **Solución 4** | | |
| x4 | y4 | z4 |
| 5 | 12 | 13 |

**Tabla 5. Soluciones analíticas al ejemplo 5.2**

Las triplas pitagóricas que cumplen son: 6, 8, 10; 12, 5, 13; 8, 6, 10; 5, 12, 13.

***Solución con el Algoritmo:***

| Ej. | Solución 1 | | | Solución 2 | | | Solución 3 | | | Solución 4 | | | Duración (s) | # Iteración |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | x1 | y1 | z1 | x2 | y2 | z2 | x3 | y3 | z3 | x4 | y4 | z4 | | |
| 1 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 399,507 | 42196 |
| 2 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 121,812 | 14620 |
| 3 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 116,593 | 12352 |
| 4 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 17,400 | 1611 |
| 5 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 27,918 | 2913 |
| 6 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 48,548 | 5262 |
| 7 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 71,623 | 7797 |
| 8 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 16,301 | 1734 |
| 9 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 375,550 | 30704 |
| 10 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 28,332 | 2954 |
| 11 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 69,835 | 7438 |
| 12 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 12,509 | 1353 |
| 13 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 115,046 | 11337 |
| 14 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 18,434 | 1907 |
| 15 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 262,902 | 26735 |
| 16 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 79,111 | 10758 |
| 17 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 13,150 | 1931 |
| 18 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 1,622 | 276 |
| 19 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 1,371 | 237 |
| 20 | 6 | 8 | 10 | 8 | 6 | 10 | 12 | 5 | 13 | 5 | 12 | 13 | 456,206 | 56402 |
| **Promedio** | | | | | | | | | | | | | **112,689** | **12026** |

Tabla 6. Resultados en algoritmo del ejemplo 5.2 para 100 partículas, 20 ciclos

De la tabla 6 se concluye que este problema posee varias soluciones y corresponden a x1=6, y1=8 y z1=10; x2=8, y2=6 y z2=10; x3=5, y3=12 y z3=13; x4=12, y4=5 y z4=13. En la figura 8 se observan las gráficas de la duración o tiempo de cómputo y el número de iteraciones en función de los ciclos de ejecución. Se concluye de las gráficas que los ciclos de iteración y el tiempo de cómputo tienen un comportamiento similar; es decir, que si se contrasta en una gráfica (figura 9) la duración en función del número de iteraciones, se encuentra una línea de tendencia y se calcula $R^2$. En los próximos ejemplos únicamente se

realizan este tipo de gráficas, puesto que con ellas se determinan parámetros de linealidad ($R^2$ mayor que 0.999).

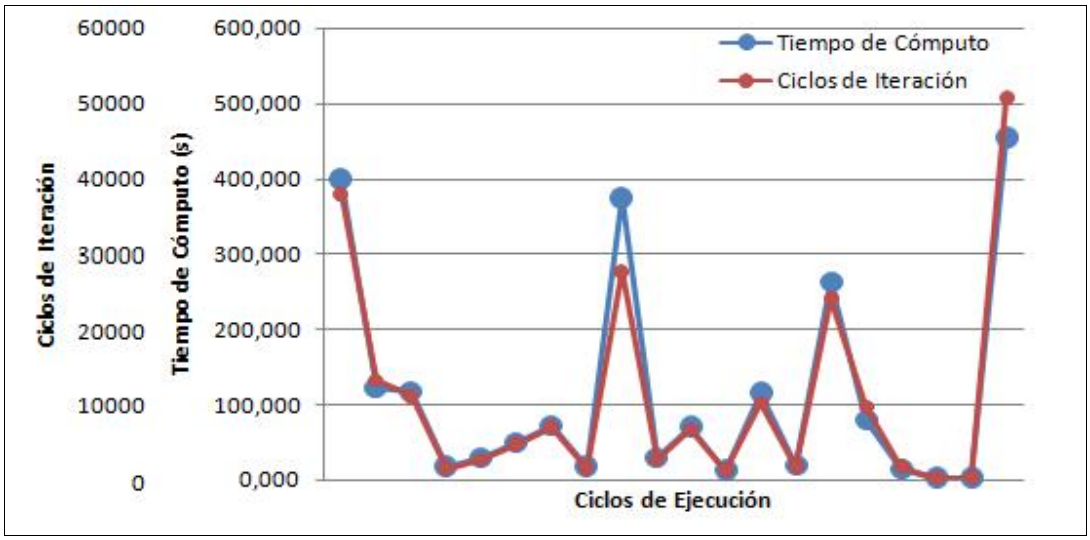


**Figura 8. Gráfica de iteraciones y tiempo de cómputo en función de ejecuciones**

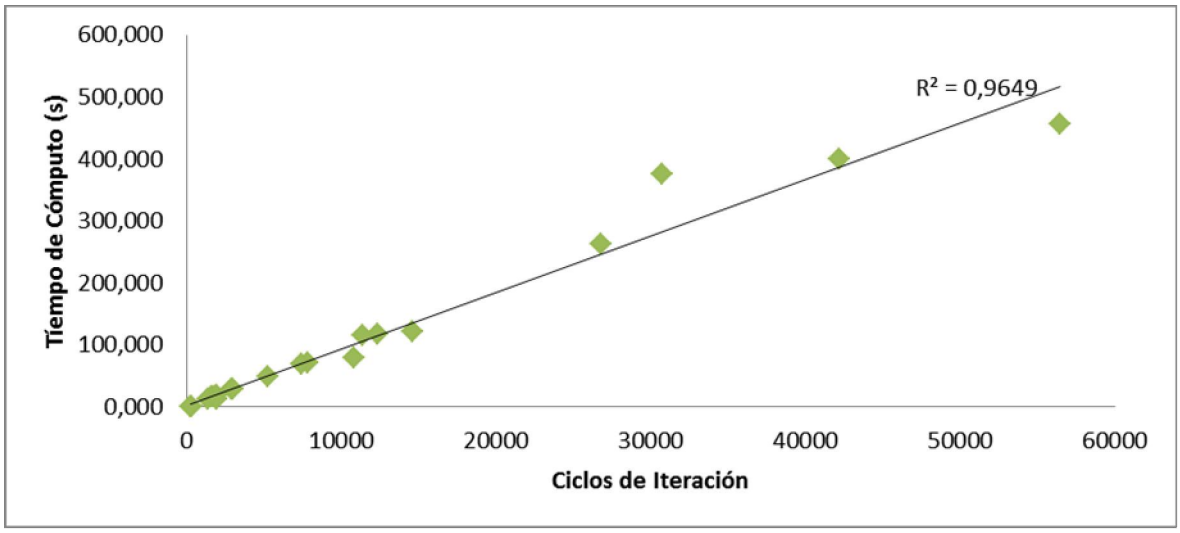


**Figura 9. Gráfica del tiempo de cómputo en función del número de ejecuciones**

## 5.3 EJEMPLO: COMPRA DE CELULARES

***Enunciado del problema:*** Una persona a través de una tienda en línea compra 12 celulares, unos de gama alta y otros de gama media, por 1200 dólares. Si los de gama alta valen 30 dólares más por unidad y ha comprado el mínimo posible de gama media, ¿Cuántos compró de cada uno? [11].

***Solución Analítica:***

Se definen $x, y, z$ como el número de teléfonos celulares de gama alta, el número celulares de gama media y el valor de los celulares de gama media respectivamente. Se plantean las ecuaciones (5.14) y (5.15). Se despeja, se resuelve y calcula encontrando las ecuaciones (5.16), (5.17), (5.18).

$$x + y = 12 \qquad (5.14)$$

$$x * (z + 30) + y * z = 1200 \qquad (5.15)$$

$$30x + 12z = 1200 \qquad (5.16)$$

$$x_0 = \frac{1200 * 1}{6} = 200, \quad (5.17) \qquad y_0 = \frac{1200 * (-2)}{6} = -400 \quad (5.18)$$

Se realiza el proceso mencionado y se encuentra que cualquier solución para este problema se presenta como la ecuación (5.19). Si $200 + 2 * t$ debe ser menor o igual a $12$ se plantea la ecuación (5.20). En la tabla 7 se encuentran los valores que satisfacen las ecuaciones (5.14) y (5.15). Dado que se desea comprar celulares gama media, pero en la mínima cantidad posible; la solución que se adapta a las condiciones del problema es la de la columna resaltada en color.

$$[200 + 2 * t, \ -400 - 5 * t] \quad con\ t \in \mathbf{Z}\text{-} \qquad (5.19)$$

$$200 + 2 * t \leq 12 \quad \rightarrow \quad t \leq -94 \qquad (5.20)$$

| Soluciones | | | | | | | |
|---|---|---|---|---|---|---|---|
| t | -94 | -95 | -96 | -97 | -98 | -99 | -100 |
| x | 12 | 10 | 8 | 6 | 4 | 2 | 0 |
| y | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
| z | 70 | 75 | 80 | 85 | 90 | 95 | 100 |

**Tabla 7. Solución analítica al ejemplo 5.3**

***Solución con el Algoritmo:***

| Ej. | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|
| 1 | 10 | 2 | 75 | 124,400 | 12043 |
| 2 | 10 | 2 | 75 | 96,310 | 9247 |
| 3 | 10 | 2 | 75 | 107,393 | 10181 |
| 4 | 10 | 2 | 75 | 37,630 | 3797 |
| 5 | 10 | 2 | 75 | 143,075 | 15161 |
| 6 | 10 | 2 | 75 | 161,560 | 15036 |
| 7 | 10 | 2 | 75 | 198,713 | 20896 |
| 8 | 10 | 2 | 75 | 63,577 | 6521 |
| 9 | 10 | 2 | 75 | 92,240 | 8387 |
| 10 | 10 | 2 | 75 | 72,660 | 7501 |
| 11 | 10 | 2 | 75 | 90,104 | 9179 |
| 12 | 10 | 2 | 75 | 148,940 | 14440 |
| 13 | 10 | 2 | 75 | 161,595 | 14831 |
| 14 | 10 | 2 | 75 | 237,327 | 22420 |
| 15 | 10 | 2 | 75 | 48,352 | 4548 |
| 16 | 10 | 2 | 75 | 205,056 | 19700 |
| 17 | 10 | 2 | 75 | 76,978 | 7243 |
| 18 | 10 | 2 | 75 | 251,780 | 24291 |
| 19 | 10 | 2 | 75 | 190,708 | 18262 |
| 20 | 10 | 2 | 75 | 262,125 | 25233 |
| **Promedio** | | | | **138,526** | **13446** |

**Tabla 8. Resultados en algoritmo del ejemplo 5.3 para 100 partículas, 20 ciclos**

De la tabla 8 se concluye que el resultado es $x = 10, \quad y = 2, \quad z = 75$; se menciona que este ejemplo posee múltiples soluciones, pero únicamente la

mencionada garantiza que los celulares de gama media comprados sean mínimos. Así mismo, en la figura 10 se observa el parámetro de linealidad $R^2$ en la gráfica.

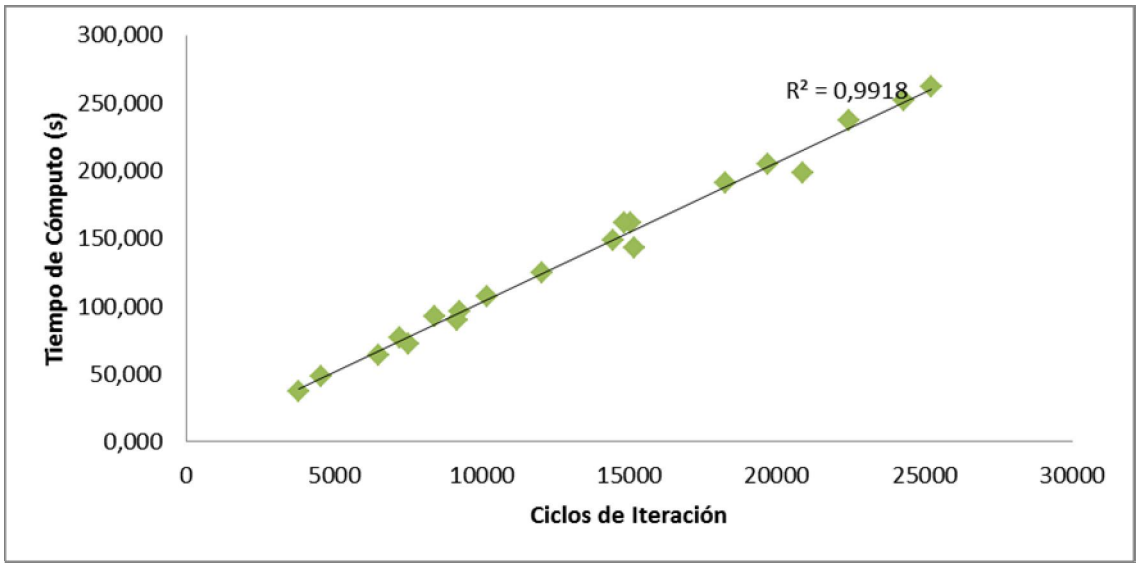


**Figura 10. Gráfica del tiempo en función del número de ejecuciones**

## 5.4 EJEMPLO: COMPRA DE INTERIORES

***Enunciado del problema:*** Un hombre compró interiores largos y cortos por 99 dólares. Si un interior largo cuesta 3 dólares más que uno corto, y compró más interiores largos que cortos, ¿Cuántos interiores compró de cada modelo? [27].

***Respuesta Analítica:***

Llamamos $x, y$ al número de interiores largos y cortos, respectivamente, $z$ al precio de un interior corto. Se plantean las ecuaciones (5.21) y (5.22):

$$x + y = 12 \qquad (5.21)$$

$$(z + 3) * x + z * y = 99 \qquad (5.22)$$

| Solución | | |
|---|---|---|
| x | y | z |
| 9 | 3 | 6 |

**Tabla 9. Respuesta analítica al ejemplo 5.4**

Por lo tanto, se compraron 9 interiores largos, 3 interiores cortos, el precio de los interiores largos es de 9 dólares y el de los interiores cortos es de 6 dólares [27].

***Solución con el Algoritmo:***

| Ej. | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|
| 1 | 9 | 3 | 6 | 72,536 | 7416 |
| 2 | 9 | 3 | 6 | 26,599 | 2820 |
| 3 | 9 | 3 | 6 | 87,695 | 9153 |
| 4 | 9 | 3 | 6 | 21,454 | 2031 |
| 5 | 9 | 3 | 6 | 119,730 | 11667 |
| 6 | 9 | 3 | 6 | 53,682 | 5707 |
| 7 | 9 | 3 | 6 | 20,797 | 2183 |
| 8 | 9 | 3 | 6 | 20,288 | 2226 |
| 9 | 9 | 3 | 6 | 14,848 | 1608 |
| 10 | 9 | 3 | 6 | 120,772 | 12751 |
| 11 | 9 | 3 | 6 | 19,493 | 2064 |
| 12 | 9 | 3 | 6 | 6,277 | 671 |
| 13 | 9 | 3 | 6 | 15,452 | 1580 |
| 14 | 9 | 3 | 6 | 19,906 | 1832 |
| 15 | 9 | 3 | 6 | 18,142 | 1636 |
| 16 | 9 | 3 | 6 | 17,524 | 1598 |
| 17 | 9 | 3 | 6 | 18,212 | 1845 |
| 18 | 9 | 3 | 6 | 187,121 | 18670 |
| 19 | 9 | 3 | 6 | 17,820 | 1752 |
| 20 | 9 | 3 | 6 | 139,490 | 13072 |
| **Promedio** | | | | **50,892** | **5114** |

**Tabla 10. Resultados en algoritmo del ejemplo 5.4 para 100 partículas**

De la tabla 10, se concluye que el resultado es $x = 9, \quad y = 3, \quad z = 6$; en la última fila se observa un promedio del tiempo de cómputo y el número de ciclos de ejecución. Para todos los ejercicios de este capítulo se ilustran estos valores promedio en la tabla de resultados. Se menciona que en este ejemplo se cumplen muchas posibles soluciones, pero únicamente la ilustrada en la tabla garantiza que se compren mayor número de interiores largos que cortos. En la figura 11, se observa la gráfica de duración vs iteraciones y el correspondiente parámetro de linealidad $R^2$ para este ejemplo.

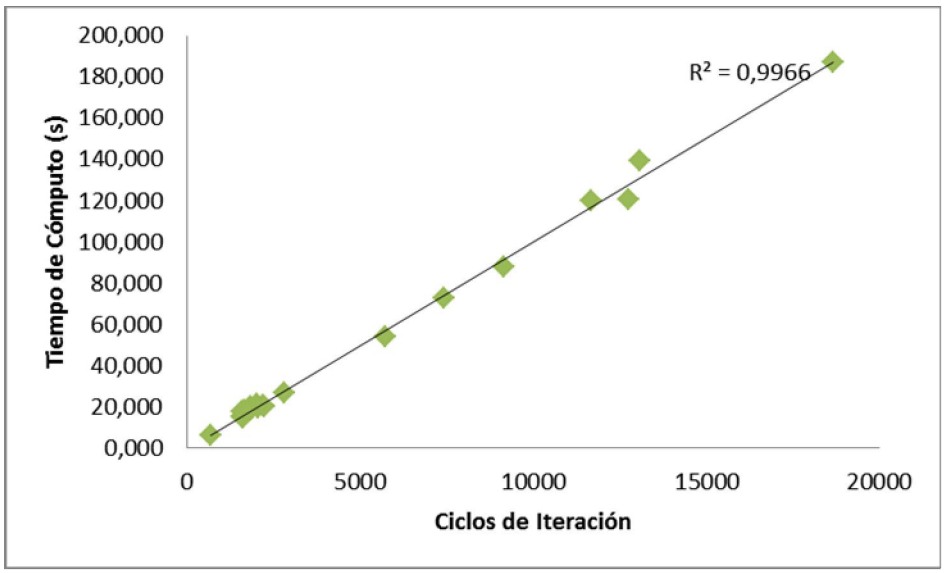


**Figura 11. Gráfica del tiempo en función del número de ejecuciones**

## 5.5 EJEMPLO: EL CHEQUE DESCONOCIDO

***Enunciado del problema:*** Un hombre cobra un cheque por $d$ dólares y $c$ centavos en un banco. El cajero por error le da $c$ dólares y $d$ centavos. El hombre no se da cuenta hasta que gasta 23 centavos, además observa que en ese momento tiene $2 * d$ dólares, $2 * c$ centavos. ¿Cuál era el valor del cheque? [27].

***Respuesta Analítica* [27]:**

| c | d |
|---|---|
| 51 | 25 |

**Tabla 11. Solución analítica al ejemplo 5.5**

Por tanto el valor del cheque era de 25 dólares 51 centavos [27].

***Solución con el Algoritmo:***

| Ej. | c | d | Duración (s) | # Iteración |
|---|---|---|---|---|
| 1 | 51 | 25 | 0,981 | 213 |
| 2 | 51 | 25 | 12,020 | 2819 |
| 3 | 51 | 25 | 0,483 | 107 |
| 4 | 51 | 25 | 4,244 | 993 |
| 5 | 51 | 25 | 38,075 | 8233 |
| 6 | 51 | 25 | 1,170 | 251 |
| 7 | 51 | 25 | 13,252 | 2884 |
| 8 | 51 | 25 | 49,004 | 9788 |
| 9 | 51 | 25 | 17,357 | 3484 |
| 10 | 51 | 25 | 3,167 | 742 |
| 11 | 51 | 25 | 7,488 | 1761 |
| 12 | 51 | 25 | 1,341 | 299 |
| 13 | 51 | 25 | 1,997 | 430 |
| 14 | 51 | 25 | 23,037 | 4710 |
| 15 | 51 | 25 | 0,764 | 160 |
| 16 | 51 | 25 | 0,405 | 85 |
| 17 | 51 | 25 | 36,108 | 7083 |
| 18 | 51 | 25 | 2,106 | 515 |
| 19 | 51 | 25 | 38,839 | 7495 |
| 20 | 51 | 25 | 47,669 | 10480 |
| **Promedio** | | | **14,975** | **3127** |

**Tabla 12. Resultados en algoritmo del ejemplo 5.5 para 100 partículas**

De la tabla 12, se concluye que el resultado es $c = 51$ y $d = 25$, nuevamente existen múltiples posibles soluciones, pero únicamente la mencionada garantiza que los centavos no sean mayor a 100, es el máximo permitido. En la figura 12, se observa la gráfica de duración vs iteraciones y el correspondiente parámetro de linealidad $R^2$ para este ejemplo.

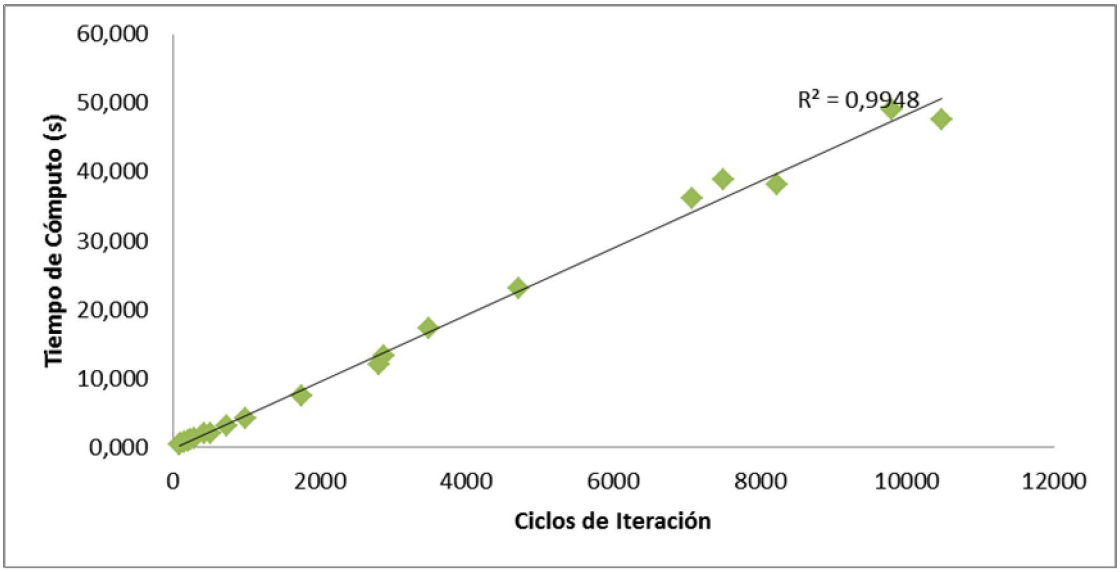


**Figura 12. Gráfica del tiempo en función del número de ejecuciones ejemplo 5.5**

## 5.6 EJEMPLO: EL GRANJERO Y SUS ANIMALES

***Enunciado del problema:*** Un granjero gastó 10.000.000 de pesos, en 100 animales entre pollos, marranos y terneros. Si compró los pollos a 5.000 pesos, los marranos a 100.000 pesos y los terneros a 500.000 pesos, y si adquirió animales de las tres clases, ¿Cuántos compró de cada uno? [28]. Sean $x, y, z$ el número de pollos, marranos y terneros respectivamente.

De acuerdo al enunciado se tiene el sistema de ecuaciones de (5.23) y (5.24). Simplificando (5.24) se obtiene la ecuación (5.25).

$$x + y + z = 100 \quad (5.23)$$

$$5000x + 100000y + 500000z = 10000000 \quad (5.24)$$

$$x + 20y + 100z = 2000 \quad (5.25)$$

***Respuesta Analítica:***

| Solución | | |
|---|---|---|
| x | y | z |
| 80 | 1 | 19 |

**Tabla 13. Solución analítica al ejemplo 5.6**

Por lo tanto compró 80 pollos, 1 marrano y 19 terneros [28].

***Solución con el Algoritmo:***

| Ej. | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|
| 1 | 80 | 1 | 19 | 98,925 | 17847 |
| 2 | 80 | 1 | 19 | 7,843 | 1175 |
| 3 | 80 | 1 | 19 | 1,186 | 204 |
| 4 | 80 | 1 | 19 | 0,905 | 152 |
| 5 | 80 | 1 | 19 | 107,596 | 18261 |
| 6 | 80 | 1 | 19 | 322,257 | 48891 |
| 7 | 80 | 1 | 19 | 3,541 | 570 |
| 8 | 80 | 1 | 19 | 78,513 | 11106 |
| 9 | 80 | 1 | 19 | 425,356 | 58965 |
| 10 | 80 | 1 | 19 | 1,409 | 199 |
| 11 | 80 | 1 | 19 | 1,210 | 222 |
| 12 | 80 | 1 | 19 | 474,043 | 66832 |
| 13 | 80 | 1 | 19 | 79,111 | 11758 |

| | | | | | |
|---|---|---|---|---|---|
| 14 | 80 | 1 | 19 | 13,150 | 1931 |
| 15 | 80 | 1 | 19 | 1,622 | 276 |
| 16 | 80 | 1 | 19 | 1,271 | 237 |
| 17 | 80 | 1 | 19 | 456,206 | 62402 |
| 18 | 80 | 1 | 19 | 1,858 | 320 |
| 19 | 80 | 1 | 19 | 103,324 | 18034 |
| 20 | 80 | 1 | 19 | 1,192 | 201 |
| **Promedio** | | | | **109,026** | **15979** |

**Tabla 14. Resultados en algoritmo del ejemplo 5.6 para 100 partículas**

De la tabla 14 se concluye que el resultado es $x = 80,\ y = 1,\ z = 19$. Los valores promedio son: duración = 109,026(s) e iteraciones = 15979. En la figura 13 se puede concluir un comportamiento lineal porque $R^2$ es igual a 0.995.

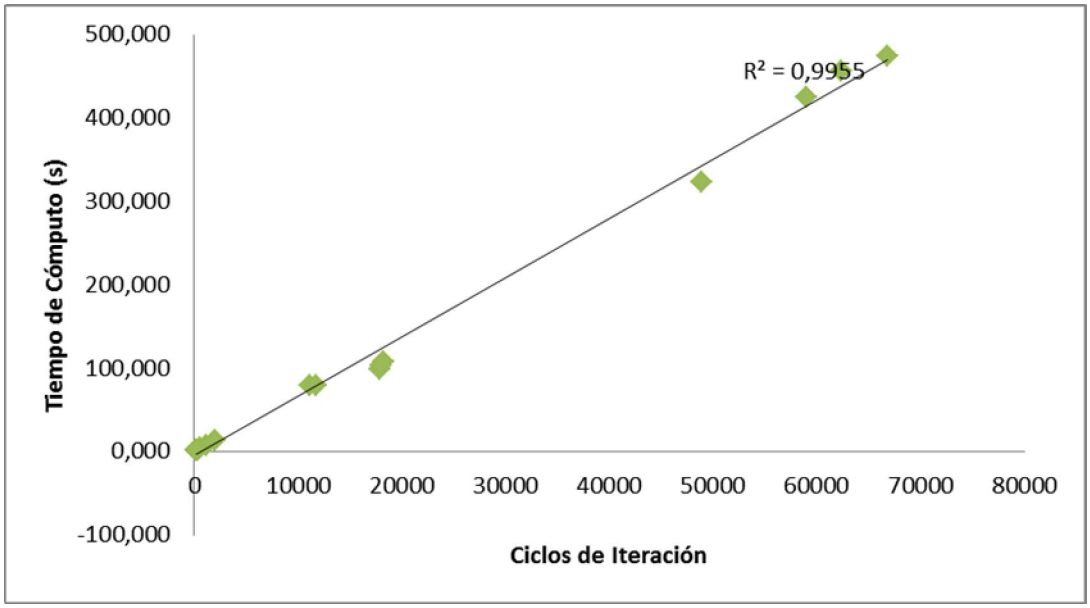


**Figura 13. Gráfica del tiempo en función del número de ejecuciones**

## 5.7 EJEMPLO: OPTIMIZACION DEL TRABAJO DE UN SASTRE

***Enunciado del problema:*** Un sastre invierte 13 horas en diseñar un modelo de pantalón y 37 horas en diseñar un modelo de camisa. Si trabaja 2.000 horas ¿Cuántas camisas y pantalones se deberán diseñar para conseguir la mayor combinación entre camisas y pantalones? [7]. Sean $x$ el número de pantalones que se diseñan, $y$ el número de camisas. Según el tiempo que le cuesta realizar cada diseño y el total de horas que invierte, se debe cumplir la ecuación diofántica (5.26).

$$13x + 37y = 2000 \quad (5.26)$$

***Respuesta Analítica:***

| Solución | | |
|---|---|---|
| x | y | (x*y) |
| 3 | 53 | 159 |
| 40 | 40 | 1600 |
| 77 | 27 | 2079 |
| 114 | 14 | 1596 |
| 151 | 1 | 151 |

**Tabla 15. Solución analítica al ejemplo 5.7**

De la tabla 15, se observa que para este problema existen múltiples soluciones, pero únicamente la resaltada (77 camisas y 27 pantalones) garantiza una combinación, que permite la producción máxima del sastre en las 2000 horas [7].

***Solución con el Algoritmo:***

| Ej. | x | y | Duración (s) | # Iteración |
|---|---|---|---|---|
| 1 | 77 | 27 | 40,010 | 4739 |
| 2 | 77 | 27 | 27,770 | 3366 |
| 3 | 77 | 27 | 78,286 | 9476 |
| 4 | 77 | 27 | 53,647 | 7199 |
| 5 | 77 | 27 | 33,559 | 4740 |
| 6 | 77 | 27 | 78,120 | 10109 |
| 7 | 77 | 27 | 25,527 | 3272 |
| 8 | 77 | 27 | 38,156 | 4782 |
| 9 | 77 | 27 | 56,873 | 7262 |
| 10 | 77 | 27 | 12,620 | 1552 |
| 11 | 77 | 27 | 70,414 | 8648 |
| 12 | 77 | 27 | 65,620 | 8411 |
| 13 | 77 | 27 | 11,582 | 1383 |
| 14 | 77 | 27 | 67,081 | 8862 |
| 15 | 77 | 27 | 97,721 | 11441 |
| 16 | 77 | 27 | 16,680 | 2217 |
| 17 | 77 | 27 | 35,530 | 5246 |
| 18 | 77 | 27 | 51,732 | 6814 |
| 19 | 77 | 27 | 53,619 | 7530 |
| 20 | 77 | 27 | 16,854 | 2309 |
| **Promedio** | | | **46,570** | **5968** |

**Tabla 16. Resultados en algoritmo del ejemplo 5.7 para 100 partículas**

De la tabla 16 se concluye que el resultado es $x = 77, y = 27$. Los valores promedio son 46,570(s) y 5968 iteraciones. En la figura 14 se observa que $R^2$ es igual a 0.984.

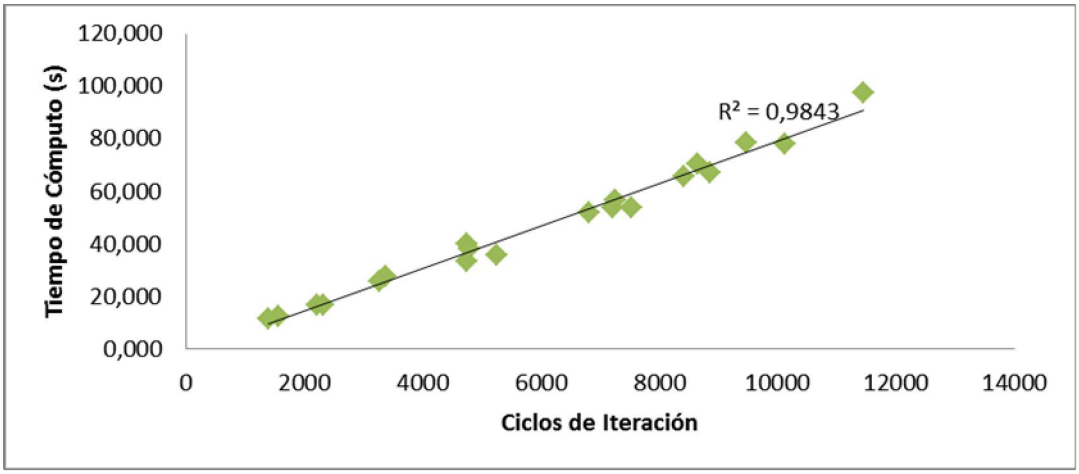


**Figura 14. Gráfica del tiempo en función del número de ejecuciones**

## 5.8 EJEMPLO: GANANCIAS EN RESTAURANTE

***Enunciado del problema:*** En un local se sirven tres tipos de menú. Un día se sirven 8, 16 y 14 menús, respectivamente y la recaudación aumenta en 15 dólares. Sabiendo que el precio de los menús es múltiplo de 0.5 dólares. ¿Se puede deducir cual es el precio? [7].

***Respuesta Analítica:***

Si llamamos $u$ al precio del menú del primer tipo, $v$ al precio del menú del segundo tipo y $w$ al precio del menú del tercer tipo, el problema consiste en resolver el sistema de ecuaciones (5.27) y (5.28). Si $u, v, w$ son múltiplos de 0.5, por lo tanto, $x = 2 * u$, $y = 2 * v$, $z = 2 * w$ y se plantea el sistema de ecuaciones diofántica (5.29) y (5.30).

$$10u + 15v + 12w = 294 \quad (5.27)$$

$$8u + 16v + 14w = 309 \quad (5.28)$$

$$10x + 15y + 12z = 588 \quad (5.29)$$

$$4x + 8y + 7z = 309 \qquad (5.30)$$

En la tabla 17 se observa que la única solución posible [7] es $x = 12$, $y = 16$ y $z = 19$. Por lo tanto los precios de los menús son $u = 6$ dólares, $v = 8$ dólares, $w = 9{,}5$ dólares [7].

| Solución | | |
|---|---|---|
| x | y | z |
| 12 | 16 | 19 |

**Tabla 17. Solución analítica al ejemplo 5.8**

***Solución con el Algoritmo:***

| Ej. | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|
| 1 | 12 | 16 | 19 | 43,340 | 5823 |
| 2 | 12 | 16 | 19 | 3,179 | 458 |
| 3 | 12 | 16 | 19 | 244,857 | 34307 |
| 4 | 12 | 16 | 19 | 119,822 | 19875 |
| 5 | 12 | 16 | 19 | 1,335 | 287 |
| 6 | 12 | 16 | 19 | 28,301 | 4095 |
| 7 | 12 | 16 | 19 | 397,196 | 43816 |
| 8 | 12 | 16 | 19 | 42,499 | 6181 |
| 9 | 12 | 16 | 19 | 35,474 | 6468 |
| 10 | 12 | 16 | 19 | 28,920 | 5086 |
| 11 | 12 | 16 | 19 | 70,434 | 12852 |
| 12 | 12 | 16 | 19 | 29,812 | 5456 |
| 13 | 12 | 16 | 19 | 120,073 | 22003 |
| 14 | 12 | 16 | 19 | 81,214 | 14755 |
| 15 | 12 | 16 | 19 | 108,249 | 19798 |
| 16 | 12 | 16 | 19 | 0,826 | 142 |
| 17 | 12 | 16 | 19 | 46,675 | 8419 |
| 18 | 12 | 16 | 19 | 125,084 | 22103 |
| 19 | 12 | 16 | 19 | 173,675 | 31667 |
| 20 | 12 | 16 | 19 | 1,934 | 349 |
| **Promedio** | | | | **85,145** | **13197** |

**Tabla 18. Resultados en algoritmo del ejemplo 5.8 para 100 partículas**

De la tabla 18 se concluye que el resultado es $x = 12$, $y = 16$, $z = 19$. Los valores promedio son 85,145(s) y 13197 iteraciones. En la figura 15 se observa que $R^2$ es igual a 0.925. En la figura 16 se gráfica una de las ejecuciones del algoritmo en Matlab.

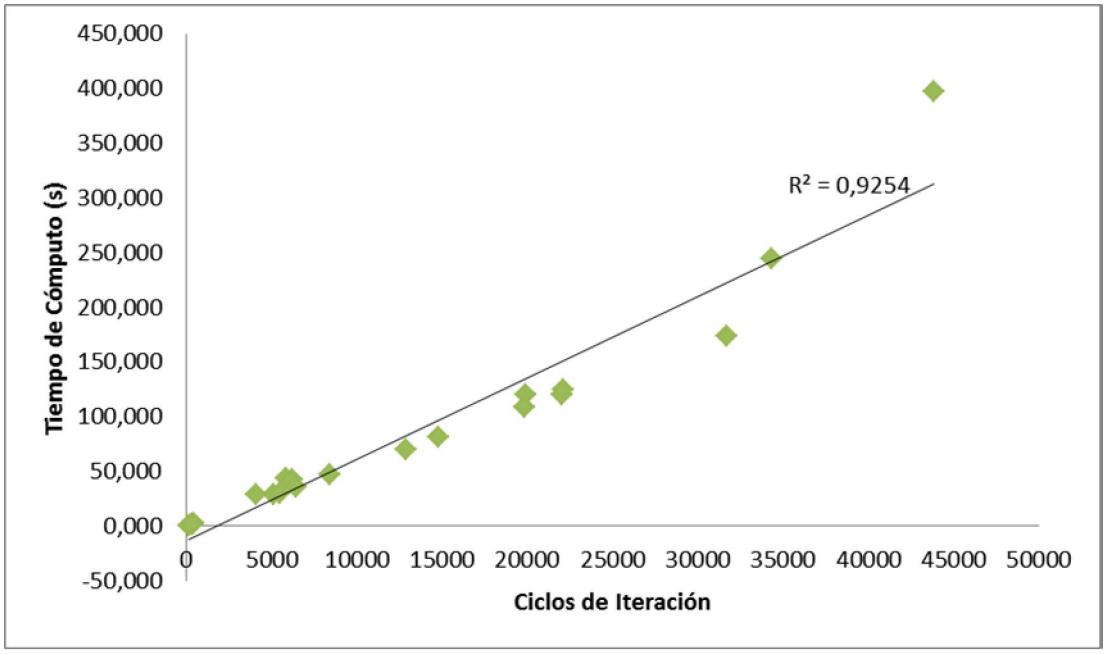


**Figura 15. Gráfica del tiempo en función del número de ejecuciones**

En la simulación (figura 16), se observa que el punto verde (partículas) está encima del área roja (solución), como el criterio de parada para el algoritmo es cuando todas las partículas sean iguales a la solución, no existe posibilidad de una partícula en otro valor.

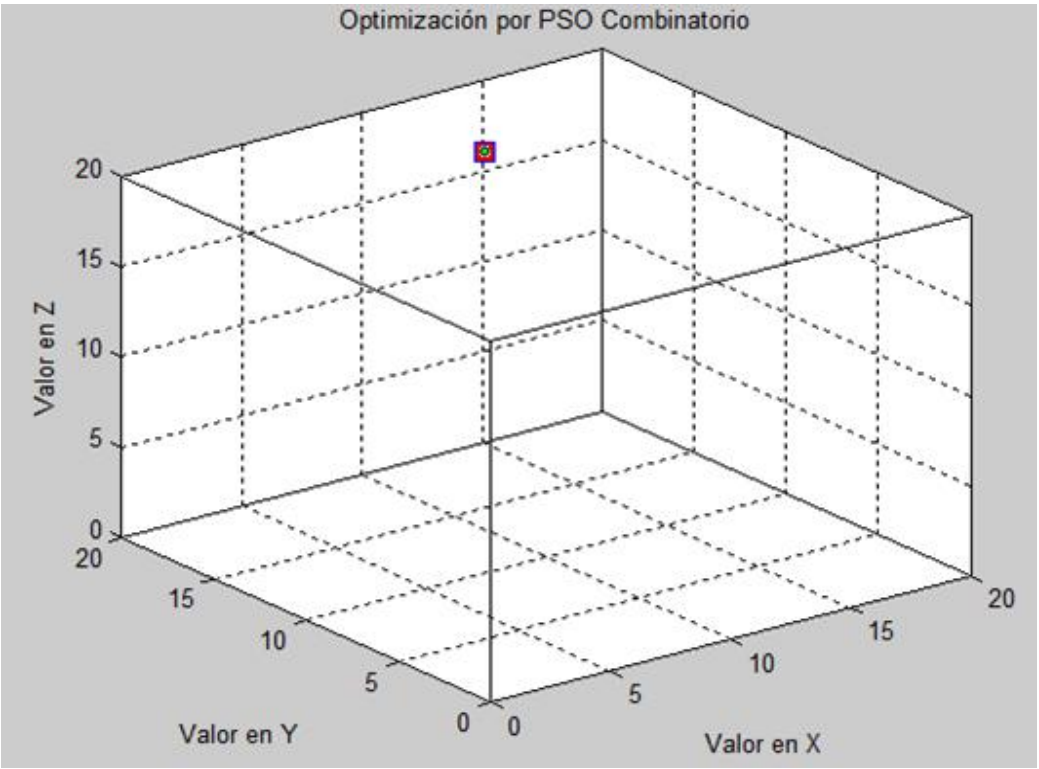


**Figura 16. Simulación del ejemplo 5.8 en el algoritmo**

## 5.9 EJEMPLO DE DISEÑO DE MECANISMOS

***Enunciado del problema:*** Se necesitan dos ruedas dentadas: Una de 237 dientes y otra de 295. No disponemos más que de ruedas de 10, 15, 20, 25, 30, 35, y 40 dientes. Determinar los dientes de cada una de ellas para efectuar el trabajo [3].

***Respuesta Analítica:***

Las ruedas utilizadas son de 25 y 20 dientes, puesto que, son múltiplos de 5. Las soluciones son $x = 5$, $y = 4$ respectivamente [3].

| Solución | |
|---|---|
| x | y |
| 5 | 4 |

**Tabla 19. Solución analítica al ejemplo 5.9**

***Solución con el Algoritmo:***

| Ej. | x | y | Duración (s) | # Iteración |
|---|---|---|---|---|
| 1 | 5 | 4 | 2,159 | 336 |
| 2 | 5 | 4 | 4,747 | 757 |
| 3 | 5 | 4 | 28,550 | 4597 |
| 4 | 5 | 4 | 42,810 | 6876 |
| 5 | 5 | 4 | 1,382 | 214 |
| 6 | 5 | 4 | 0,581 | 87 |
| 7 | 5 | 4 | 18,880 | 2996 |
| 8 | 5 | 4 | 3,718 | 588 |
| 9 | 5 | 4 | 1,008 | 153 |
| 10 | 5 | 4 | 3,141 | 501 |
| 11 | 5 | 4 | 0,792 | 118 |
| 12 | 5 | 4 | 2,313 | 358 |
| 13 | 5 | 4 | 3,304 | 517 |
| 14 | 5 | 4 | 6,002 | 943 |
| 15 | 5 | 4 | 24,290 | 3900 |
| 16 | 5 | 4 | 0,716 | 106 |
| 17 | 5 | 4 | 1,470 | 228 |
| 18 | 5 | 4 | 25,030 | 4038 |
| 19 | 5 | 4 | 0,958 | 142 |
| 20 | 5 | 4 | 12,870 | 2068 |
| **Promedio** | | | **9,236** | **1476** |

**Tabla 20. Resultados en algoritmo del ejemplo 5.9 para 100 partículas**

De la tabla 20 se concluye que el resultado es $x = 5$, $y = 4$. Los valores promedio son 9,236(s) y 1476 iteraciones. En la figura 17 se evidencia un comportamiento lineal porque $R^2$ es igual a 1.

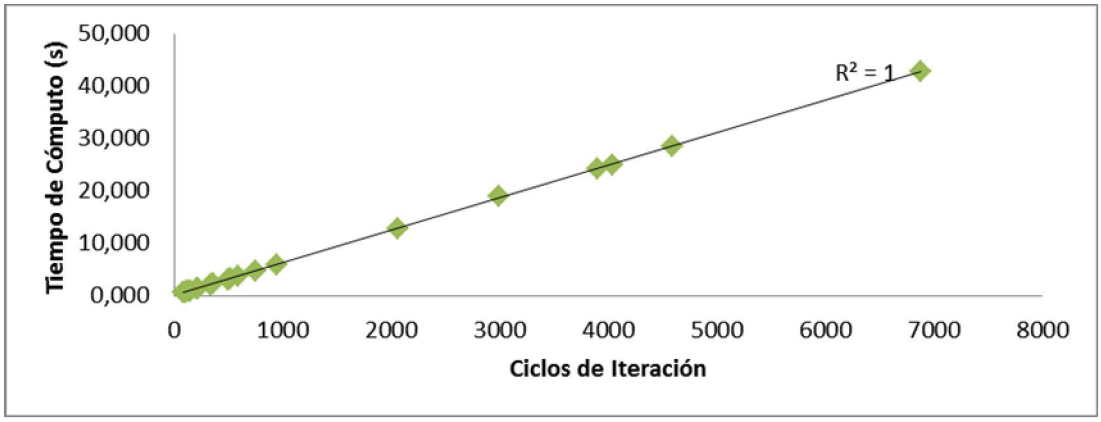


**Figura 17. Gráfica del tiempo en función del número de ejecuciones**

## 5.10 EJEMPLO EN PROCESOS DE FABRICACIÓN

***Enunciado del problema:*** En una empresa de textiles se tienen seis máquinas que permiten confeccionar diferentes tipos de prendas (blusas, camisas, pantalones, chaquetas, toallas, fundas), si cada una de ellas gasta un determinado tiempo y tiene un gasto en mantenimiento, calcular cuantas prendas se pueden confeccionar al día. Las ecuaciones que modelan el sistema son (5.31), (5.32), (5.33), (5.34), (5.35), (5.36), [22].

$$5u + 10v - 5w + y^3 + 8z = 1772 \qquad (5.31)$$

$$3u + 18w - 5y + 17z = 153 \qquad (5.32)$$

$$6u + w - y + (15z)^2 = 1772 \qquad (5.33)$$

$$-u + 5v + 8w - 6x + 15y + 10z = 277 \qquad (5.34)$$

$$(u + v)^2 - 7w + 5x + 12y - 8z = 150 \qquad (5.35)$$

$$v - 5w + 3y - z = -4 \qquad (5.36)$$

***Respuesta por Simulación:***

El anterior problema fue modelado en el software Arena [22] y encontró como respuesta los valores de la tabla 21.

| Solución | | | | | |
|---|---|---|---|---|---|
| u | v | w | x | y | z |
| 6 | 3 | 8 | 1 | 12 | 3 |

**Tabla 21. Solución analítica al ejemplo 5.10**

De la tabla 21 se puede decir que se pueden confeccionar al día 6 blusas, 3 camisas, 8 pantalones, 1 chaqueta, 12 toallas, 3 fundas [22].

***Solución con el Algoritmo:***

| Ej. | u | v | w | x | y | z | Duración (s) | # Iteración |
|---|---|---|---|---|---|---|---|---|
| 1 | 6 | 3 | 8 | 1 | 12 | 3 | 9,556 | 507 |
| 2 | 6 | 3 | 8 | 1 | 12 | 3 | 9,135 | 512 |
| 3 | 6 | 3 | 8 | 1 | 12 | 3 | 16,560 | 922 |
| 4 | 6 | 3 | 8 | 1 | 12 | 3 | 9,376 | 519 |
| 5 | 6 | 3 | 8 | 1 | 12 | 3 | 25,843 | 1438 |
| 6 | 6 | 3 | 8 | 1 | 12 | 3 | 14,218 | 789 |
| 7 | 6 | 3 | 8 | 1 | 12 | 3 | 10,374 | 584 |
| 8 | 6 | 3 | 8 | 1 | 12 | 3 | 70,633 | 3951 |
| 9 | 6 | 3 | 8 | 1 | 12 | 3 | 5,499 | 307 |
| 10 | 6 | 3 | 8 | 1 | 12 | 3 | 9,060 | 504 |
| 11 | 6 | 3 | 8 | 1 | 12 | 3 | 31,419 | 1775 |
| 12 | 6 | 3 | 8 | 1 | 12 | 3 | 19,178 | 1080 |
| 13 | 6 | 3 | 8 | 1 | 12 | 3 | 11,467 | 644 |
| 14 | 6 | 3 | 8 | 1 | 12 | 3 | 10,911 | 614 |
| 15 | 6 | 3 | 8 | 1 | 12 | 3 | 47,570 | 2676 |
| 16 | 6 | 3 | 8 | 1 | 12 | 3 | 5,138 | 282 |
| 17 | 6 | 3 | 8 | 1 | 12 | 3 | 9,166 | 544 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 18 | 6 | 3 | 8 | 1 | 12 | 3 | 18,499 | 1026 |
| 19 | 6 | 3 | 8 | 1 | 12 | 3 | 9,991 | 564 |
| 20 | 6 | 3 | 8 | 1 | 12 | 3 | 23,252 | 1265 |
| **Promedio** | | | | | | | **18,342** | **1025** |

**Tabla 22. Resultados en algoritmo del ejemplo 5.10 para 100 partículas**

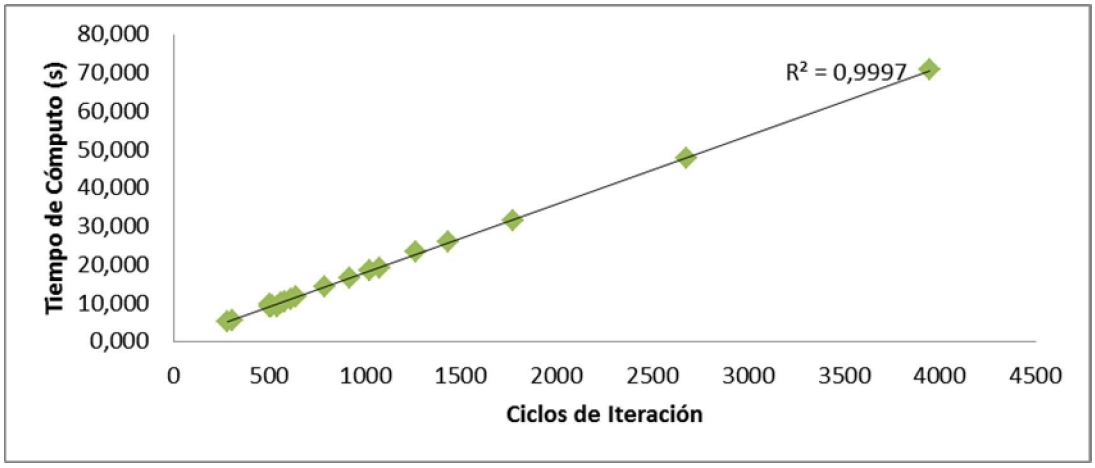


**Figura 18. Gráfica del tiempo en función del número de ejecuciones**

En la tabla 22 se encuentra que el resultado es $u = 6,\ v = 3,\ w = 8,\ x = 1,\ y = 12,\ z = 3$. Los valores promedio son 18,342(s) y 1025 iteraciones. En la figura 18 se observa un comportamiento lineal porque $R^2$ es igual a 0.999.

# 6. APLICACIÓN EN INGENIERÍA ELECTRÓNICA

Esta sección se desarrolla con fin de cumplir el objetivo del trabajo de grado:

- Implementar una estrategia de optimización combinatoria con enjambre de partículas CPSO para resolver problemas de ingeniería electrónica donde se presenten ecuaciones diofánticas.

Las ecuaciones diofánticas son utilizadas para solucionar algunos problemas de ingeniería electrónica. En la teoría de control, donde las matrices polinómicas juegan un papel importante (ya que describen el comportamiento de un sistema), la utilización de las ecuaciones diofánticas canonícas es importante, puesto que, modelan el problema y se pueden resolver, a través del algoritmo implementado en este trabajo de grado. En [5], se solucionan algunos problemas de control, primero con un modelo en el dominio del tiempo en espacio de estados, segundo en el dominio de la frecuencia usando funciones de transferencia y tercero, con polinomios y la matriz de Silvester.

En el diseño de controladores con retardos en el tiempo [16], mediante la ubicación de los polos y ceros, existe una propuesta alternativa para dos casos de polinomios característicos de lazo cerrado. El controlador obtenido es estable y cuando una planta presenta retraso, tiende a desestabilizarse. Para este tipo de problema, usualmente se utiliza un control PI ó PID. El manejo algebraico de matrices polinómicas, en sistemas de control, ha incrementado en los últimos años, permitiendo la búsqueda de nuevas técnicas para resolver problemas mediante algoritmos de optimización en control, ofreciendo buenos resultados en cuanto a estabilidad y eficiencia [14].

En un control óptimo lineal para sistemas SISO, basado en las propiedades de polinomios y solución de una ecuación diofántica lineal, una tendencia es encontrar un controlador óptimo mediante algebra polinómica. El problema se reduce en resolver ecuaciones diofánticas con polinomios. Este enfoque es importante pero no ha sido generalizado para sistemas no constantes y no lineales [19].

En control adaptativo, se encuentra que los parámetros del controlador dependen de soluciones polinómicas de ecuaciones y sus coeficientes, representan una buena respuesta siguiendo una señal de referencia. Este método también permite la sintonización de los parámetros [16]. Existe una aplicación práctica con un control simple de tiempo continuo, para la temperatura de un túnel de aire caliente en un laboratorio con parámetros de incertidumbre, utilizando controladores PI y PID sintonizados, asegurando estabilidad. Matemáticamente, este modelo se trabaja a partir de ecuaciones diofánticas, parametrización de Youla-Kucera y condiciones de divisibilidad [13]. Otra estrategia es realizar un control predictivo (MPC), el cual minimiza el espectro de los picos de error y utiliza ecuaciones diofánticas [21].

En aplicaciones para mejorar el rendimiento de una máquina, se desarrollan transformaciones con técnicas y algoritmos. El modelado geométrico con el propósito de optimizar el programa de ejecución, se aplica para el manejo por análisis de dependencia de datos [9]. Las soluciones presentadas no son polinómicas, pero dan soluciones exactas en enteros con una dependencia del valor de un parámetro. Este enfoque permite trabajar con ecuaciones diofánticas, en sistemas digitales y diseño de procesadores de múltiple propósito [18]. En la teoría de señales y comunicaciones digitales, existe una aplicación, en donde se diseña un filtro adaptativo óptimo, para mejorar una señal de voz con ruido blanco y con ruido de color, resolviendo una ecuación diofántica [12].

## 6.1 EJEMPLO ILUSTRATIVO DE INGENIERÍA ELECTRÓNICA No. 1

***Enunciado del Problema:***

Considere el sistema de control de lazo cerrado, figura 19, con realimentación unitaria. Encuentre el controlador $C(S)$, mediante el método polinómico y que satisface la ubicación de 6 polos en -1. [20] Sea la planta: $G(S) = \frac{s^2+s+1}{s^3+3s+4s+3}$

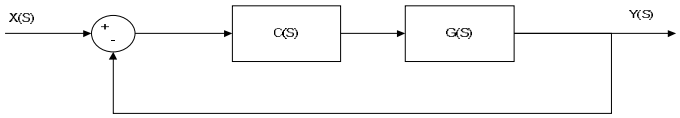


**Figura 19. Diagrama de bloques del ejemplo 6.1**

***Modelado del Problema:***

Trabajando en el dominio de la frecuencia $s$, se define la función de transferencia de lazo cerrado (6.1), $G(s)$ (6.2) y $C(s)$ (6.3). A continuación se desarrolla algebraicamente el modelo (6.4) y finalmente se plantea la ecuación diofántica (6.5).

$$\frac{Y(s)}{R(s)} = \frac{C(s) * G(s)}{1 + C(s) * G(s)} \tag{6.1}$$

$$G(s) = \frac{B(s)}{A(s)} \tag{6.2}$$

$$C(s) = \frac{N(s)}{D(s)} \tag{6.3}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{B(s)}{A(s)} * \frac{N(s)}{D(s)}}{1 + \frac{B(s)}{A(s)} * \frac{N(s)}{D(s)}} = \frac{\frac{B(s)}{A(s)} * \frac{N(s)}{D(s)}}{\frac{B(s)}{A(s)} * \frac{N(s)}{D(s)} + \frac{A(s)}{A(s)} * \frac{D(s)}{D(s)}} = \frac{B(s) * N(s)}{B(s) * N(s) + A(s) * D(s)} \tag{6.4}$$

$$D(s) * A(s) + N(s) * B(s) = F(s) \tag{6.5}$$

Si se tienen 6 polos en $s = -1$, se encuentra $F(s)$ (6.6). Con la información dada en el enunciado se conocen $A(s)$ (6.7) y $B(s)$ (6.8). En este problema se busca resolver la ecuación diofántica (6.5) y encontrar $D(s)$ (6.9) y $N(s)$ (6.10).

$$F(s) = (s+1)^6 = s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1 \quad (6.6)$$

$$A(s) = s^3 + 3s^2 + 4s + 3 \quad (6.7)$$

$$B(s) = s^2 + s + 1 \quad (6.8)$$

$$D(s) = X_1 s^3 + X_2 s^2 + X_3 s + X_4 \quad (6.9)$$

$$N(s) = X_5 s^2 + X_6 s + X_7 \quad (6.10)$$

***Solución analítica:***

La solución analítica de este problema se encuentra en [20], cuyas respuestas son:

$$X_1 = 1, \quad X_2 = 3, \quad X_3 = 2, \quad X_4 = 2, \quad X_5 = 0, \quad X_6 = -3, \quad X_7 = -5$$

Con estas respuestas se puede implementar el controlador de la ecuación (6.11).

$$C(S) = \frac{-3 * s - 5}{s^3 + 3 * s^2 + 2 * s + 2} \quad (6.11)$$

***Solución con el algoritmo:***

Se reemplazan (6.6), (6.7), (6.8), (6.9), (6.10) en la ecuación diofántica (6.5) y se obtiene (6.12).

$$(X_1 s^3 + X_2 s^2 + X_3 s + X_4) * (s^3 + 3s^2 + 4s + 3) + (X_5 s^2 + X_6 s + X_7) * (s^2 + s + 1) = s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1 \quad (6.12)$$

Se desarrolla (6.12) y se tiene (6.13)

$$(X_1) * s^6 + (3X_1 + X_2) * s^5 + (X_1 + 3X_2 + X_3 + X_5) * s^4 + (3X_1 + 4X_2 + 3X_3 + X_4 + X_5 + X_6) * s^3$$

$$+(3X_2 + 4X_3 + 3X_4 + X_5 + X_6 + X_7) * s^2 + (3X_3 + 4X_4 + X_6 + X_7) * s + (3X_4 + X_7)$$

$$= s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1 \quad (6.13)$$

De la ecuación (6.13), se plantea el sistema de ecuaciones:

$$X_1 - 1 = 0 \quad (6.14)$$

$$3X_1 + X_2 - 6 = 0 \quad (6.15)$$

$$4X_1 + 3X_2 + X_3 + X_5 - 15 = 0 \quad (6.16)$$

$$3X_1 + 4X_2 + 3X_3 + X_4 + X_5 + X_6 - 20 = 0 \quad (6.17)$$

$$3X_2 + 4X_3 + 3X_4 + X_5 + X_6 + X_7 - 15 = 0 \quad (6.18)$$

$$3X_3 + 4X_4 + X_6 + X_7 - 6 = 0 \quad (6.19)$$

$$3X_4 + X_7 - 1 = 0 \quad (6.20)$$

A continuación con las ecuaciones (6.14), (6.15), (6.16), (6.17), (6.18), (6.19), (6.20), se calcula la función objetivo (6.21).

$$F(x) = (X_1 - 1)^2 + (3X_1 + X_2 - 6)^2 + (4X_1 + 3X_2 + X_3 + X_5 - 15)^2 + (3X_1 + 4X_2 + 3X_3 + X_4 + X_5 + X_6 - 20)^2 + (3X_2 + 4X_3 + 3X_4 + X_5 + X_6 + X_7 - 15)^2 + (3X_3 + 4X_4 + X_6 + X_7 - 6)^2 + (3X_4 + X_7 - 1)^2$$

(6.21)

En la ejecución del algoritmo para los ejemplos de esta sección, se ajustaron las constantes para que el algoritmo converja rápidamente: w = 0.75, c1 = 0.9, c2 = 0.2, alfa = 0.5. En la tabla 23 se encuentran los resultados de la simulación en el algoritmo.

| Ej. | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | Duración (s) | # Iteración |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 195,755 | 11656 |
| 2 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 102,080 | 5961 |
| 3 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 227,650 | 12402 |
| 4 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 134,400 | 7753 |
| 5 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 214,700 | 12139 |
| 6 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 199,650 | 9785 |
| 7 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 240,060 | 13055 |
| 8 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 187,750 | 10441 |
| 9 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 251,080 | 12903 |
| 10 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 242,990 | 13643 |
| 11 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 199,130 | 10986 |
| 12 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 445,080 | 22805 |
| 13 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 176,770 | 8935 |
| 14 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 210,030 | 11377 |
| 15 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 187,900 | 9797 |
| 16 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 314,180 | 17330 |
| 17 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 155,070 | 8228 |
| 18 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 221,060 | 11335 |
| 19 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 266,240 | 14567 |
| 20 | 1 | 3 | 2 | 2 | 0 | -3 | -5 | 359,920 | 18366 |
| **Promedio** | | | | | | | | **226,575** | **12173** |

**Tabla 23. Resultados simulación en algoritmo Ejemplo 6.1**

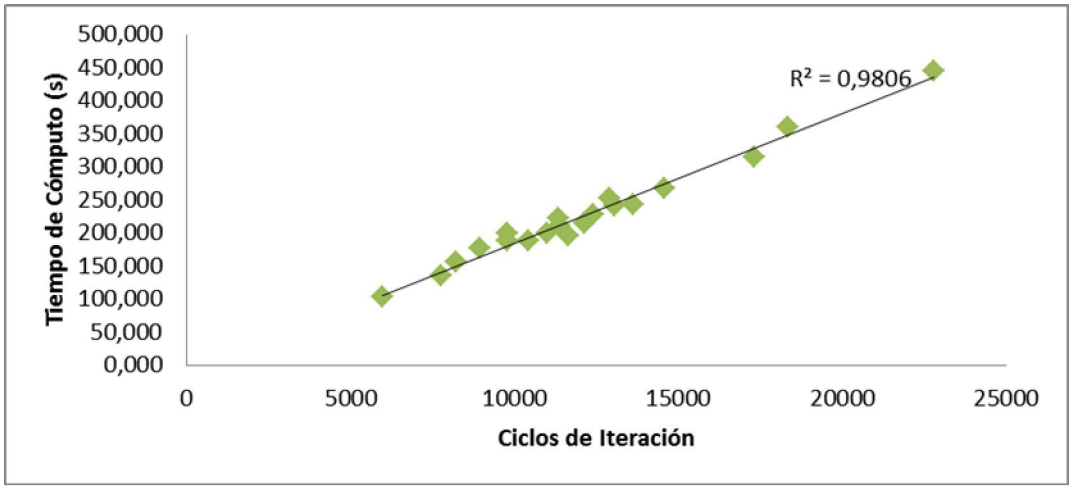


**Figura 20. Gráfica del tiempo en función del número de ejecuciones**

En la tabla 23 se observa que el resultado es:

$X_1 = 1, \quad X_2 = 3, \quad X_3 = 2, \quad X_4 2, \quad X_5 = 0, \quad X_6 = -3, \quad X_7 = -5.$

De la figura 20 se encuentra el tiempo de cómputo en función de los ciclos de iteración, $R^2$ es igual a 0.990.

## 6.2 EJEMPLO ILUSTRATIVO DE INGENIERÍA ELECTRÓNICA No. 2

***Enunciado del Problema:***

Considere el sistema de control de lazo cerrado, figura 21, con realimentación unitaria. Encuentre el controlador $C(s)$, con el método polinómico garantizando estabilidad y tiempo de establecimiento menor a 20 (s). El polo dominante de lazo cerrado esta en $s = 0{,}4016$, para la planta: $G(s) = \frac{10s+3}{(s+1)*(s+2)*(s+8)}$.

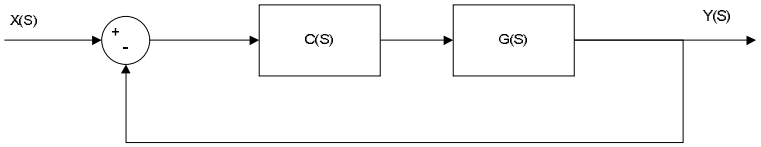


**Figura 21. Diagrama de bloques del ejemplo 6.2**

***Solución Analítica:***

La función de transferencia de lazo cerrado es (6.22). Si existe un polo dominante en $s = 0{,}4016$, entonces, $F(s) = 3s^6 + 37s^5 + 125s^4 + 196s^3 + 166s^2 + 97s + 22.$

$$\frac{Y(s)}{X(s)} = \frac{C(s)*G(s)}{1+C(s)*G(s)} \qquad (6.22)$$

En el modelado del ejercicio 6.1 se planteó y desarrolló una ecuación diofántica (6.23), se encontró solución a esta ecuación colocando $C(s)$ en función de $N(s)$ y $D(s)$ (6.24). El objetivo de este nuevo problema es encontrar los valores de $X_1, X_2, \dots, X_7$ que satisfacen las condiciones de $F(s)$. A través del algoritmo de Euclides se hallaron las soluciones de la ecuación (6.25).

$$D(s) * A(s) + N(s) * B(s) = F(s) \tag{6.23}$$

$$C(s) = \frac{X_5 s^2 + X_6 s + X_7}{X_1 s^3 + X_2 s^2 + X_3 s + X_4} \tag{6.24}$$

$$X_1 = 3, \quad X_2 = 4, \quad X_3 = 3, \quad X_4 = 1, \quad X_5 = 1, \quad X_6 = 1, \quad X_7 = 2$$

$$C(s) = \frac{s^2 + s + 2}{3s^3 + 4s^2 + 3s + 1} \tag{6.25}$$

***Solución con el Algoritmo:***

Para solucionar el ejemplo 6.2 con el algoritmo, se debe resolver la ecuación diofántica (6.23), y se plantea la ecuación (6.26).

$$\begin{aligned}(X_1 s^3 + X_2 s^2 + X_3 s + X_4) * (s^3 + 11s^2 + 26s + 16) + (X_5 s^2 + X_6 s + X_7) \\ * (10 * s + 3)\end{aligned}$$

$$= 3s^6 + 37s^5 + 125s^4 + 196s^3 + 166s^2 + 97s + 22 \tag{6.26}$$

Se desarrolla (6.26) y se obtiene (6.27).

$$\begin{aligned}(X_1) * s^6 + (11X_1 + X_2) * s^5 + (26X_1 + 11X_2 + X_3) * s^4 \\ + (16X_1 + 26X_2 + 11X_3 + X_4 + 10X_5) * s^3\end{aligned}$$

$$\begin{aligned}+(16X_2 + 26X_3 + 11X_4 + 3X_5 + 10X_6) * s^2 + (16 + 26X_4 + 3X_6 + 10X_7) * s \\ + (16X_4 + 3X_7)\end{aligned}$$

$$= 3s^6 + 37s^5 + 125s^4 + 196s^3 + 166s^2 + 97s + 22 \tag{6.27}$$

De la ecuación (6.27), se plantea el sistema de ecuaciones:

$$X_1 - 3 = 0 \quad (6.28)$$

$$11X_1 + X_2 - 37 = 0 \quad (6.29)$$

$$26X_1 + 11X_2 + X_3 - 125 = 0 \quad (6.30)$$

$$16X_1 + 26X_2 + 11X_3 + X_4 + 10X_5 - 196 = 0 \quad (6.31)$$

$$16X_2 + 26X_3 + 11X_4 + 3X_5 + 10X_6 - 166 = 0 \quad (6.32)$$

$$16X_3 + 26X_4 + 3X_6 + 10X_7 - 97 = 0 \quad (6.33)$$

$$16X_4 + 3X_7 - 22 = 0 \quad (6.34)$$

A continuación con las ecuaciones (6.28), (6.29), (6.30), (6.31), (6.32), (6.33), (6.34), se deduce la función objetivo (6.35).

$$F(x) = (X_1 - 3)^2 + (11X_1 + X_2 - 37)^2 + (26X_1 + 11X_2 + X_3 - 125)^2 + (16X_1 + 26X_2 + 11X_3 + X_4 + 10X_5 - 196)^2 + (16X_2 + 26X_3 + 11X_4 + 3X_5 + 10X_6 - 166)^2 + (16X_3 + 26X_4 + 3X_6 + 10X_7 - 97)^2 + (16X_4 + 3X_7 - 22)^2$$

(6.35)

| Ej. | X1 | X2 | X3 | X4 | X5 | X6 | X7 | Duración (s) | # Iteración |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 256,058 | 11639 |
| 2 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 365,662 | 16621 |
| 3 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 269,930 | 12315 |
| 4 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 271,678 | 12349 |
| 5 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 237,094 | 10777 |
| 6 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 251,248 | 11202 |
| 7 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 145,032 | 7001 |
| 8 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 290,208 | 14121 |
| 9 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 289,253 | 13268 |
| 10 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 153,558 | 6715 |
| 11 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 305,320 | 13974 |
| 12 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 436,083 | 19386 |
| 13 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 315,921 | 15272 |
| 14 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 214,736 | 10274 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 15 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 158,243 | 7425 |
| 16 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 266,241 | 11655 |
| 17 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 389,477 | 15595 |
| 18 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 273,460 | 12430 |
| 19 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 392,480 | 17840 |
| 20 | 3 | 4 | 3 | 1 | 1 | 1 | 2 | 343,574 | 15617 |
| **Promedio** | | | | | | | | **281,263** | **12774** |

**Tabla 24. Resultados simulación en algoritmo Ejemplo 6.2**

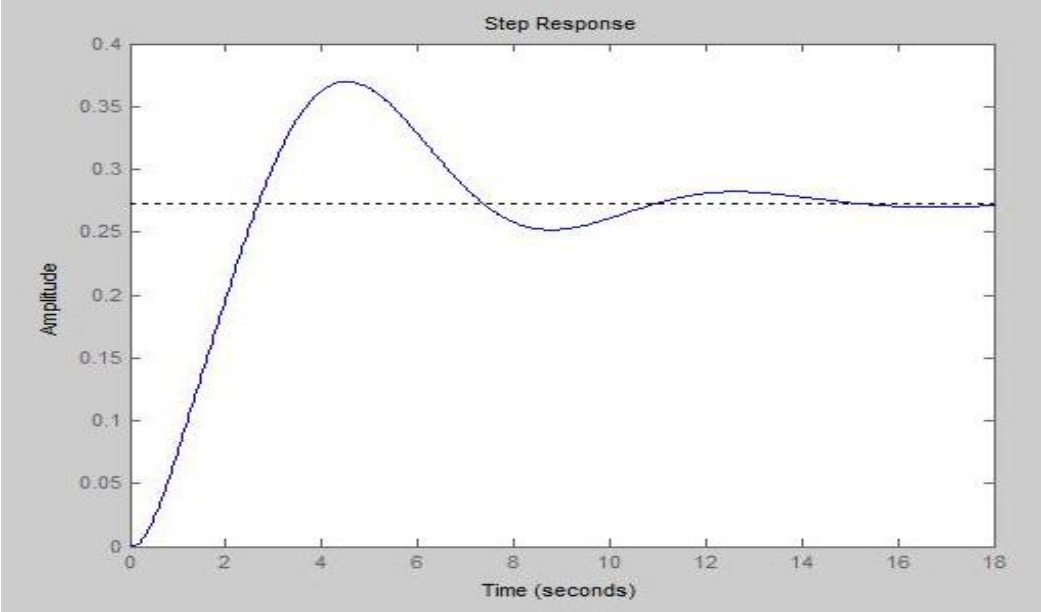


**Figura 22. Respuesta al escalón del sistema controlado**

En la tabla 24 se observa que el resultado es:

$$X_1 = 3, \quad X_2 = 4, \quad X_3 = 3, \quad X_4 = 1, \quad X_5 = 1, \quad X_6 = 1, \quad X_7 = 2.$$

El tiempo de cómputo promedio es 35,072 (s) y los ciclos de iteración promedio son 1302. En la figura 22 se tiene la respuesta al escalón de la planta con el controlador, es estable y tiene un tiempo de establecimiento menor a 10 (s) como lo requiere el ejemplo.

# 7. CONCLUSIONES

- ✓ Se comprobó la eficiencia del algoritmo, al resolver problemas de ingeniería electrónica que involucran ecuaciones diofánticas.

- ✓ Se realizó un diagrama de flujo del algoritmo, que facilitó la implementación del mismo en Matlab; lo anterior con fin de proveer una herramienta para desarrollar otros problemas, no tratados en este trabajo de grado.

- ✓ Al implementar el algoritmo de optimización combinatoria se comprobó el efecto estocástico del mismo, encontrando soluciones exactas a los problemas, que tardaron diferentes tiempos y que tomaron diferentes ciclos de cómputo. Además, para los problemas de ingeniería electrónica, se ajustaron las constantes con valores experimentales, que hacen que converja más rápido el algoritmo, reduciendo significativamente el tiempo de cómputo.

- ✓ Al resolver el ejercicio de control y encontrar las ecuaciones diofánticas, se comprobó la utilidad del algoritmo en ingeniería electrónica. Se resalta, que para encontrar la solución por medio de PSO combinatorio, no se necesitan técnicas matemáticas avanzadas, resolviendo problemas con buena calidad de resultados.

- ✓ La diferencia del algoritmo, es resolver las ecuaciones diofánticas que se presentan en algunas aplicaciones de ingeniería electrónica, sin emplear métodos algebraicos de control o de matrices.

# 8. RECOMENDACIONES

A criterio de los autores se abre la posibilidad de retomar este proyecto de grado como punto de partida para la actualización del mismo, presentando las siguientes recomendaciones:

- ✓ Mejorar el tiempo de ejecución del algoritmo a través de la optimización del código, permitiendo agilizar la obtención de resultados.

- ✓ Buscar nuevas aplicaciones en ingeniería en las que solucionar ecuaciones diofánticas sea un objetivo principal, y realizar las respectivas simulaciones de ellas en el algoritmo implementado.

- ✓ Realizar un barrido de todos los problemas para determinar estadísticamente cuáles son los valores de alfa (α), w, c1 y c2 que hacen que converja más rápidamente el algoritmo.

- ✓ Estudiar otros algoritmos evolutivos como el GSA y CFO con el fin de buscar un ajuste combinatorio de los mismos y solucionar ecuaciones diofánticas a través de la implementación de un nuevo algoritmo.

## BIBLIOGRAFÍA


[1] E. Bonilla, G. Figueria, and Malabre, “Solving the Diophantine Equation by State Space Inversion Techniques: An illustrative Example,” in *Proceedings of the 2006 American Control Conference*, 2006, vol. 1, no. 6, pp. 3731-3736.

[2] S. Abraham, S. Sugata, and M. Sanglikar, “Particle Swarm Optimization Based Diophantine Equation Solver, Journal of Bio-Inspired Computation, Volume 2 Issue 2, March 2010, url: http://arxiv.org/ftp/arxiv/papers/1003/1003.2724.pdf.

[3] A. Figueroa, “Ecuaciones diofánticas y engranajes cilíndricos,” in *Sociedad Colombiana de Matemáticas XV Congreso Nacional de Matemáticas*, 2006, vol. Especial L, pp. 285 - 298.

[4] H. Y. Cheung, “Algebraic Algorithms in Combinatorial Optimization,” The Chinese University of Hong Kong, 2011.

[5] V. Hanta, “Solution of Simple Diphantine Equations By Means of Matlab.” Ed. Prague, pp. 1 - 8, 2008.

[6] R. Fletcher, *Practical methods of optimization*, Second Edi. Scotland UK: Interscience Publication, 2000, pp. 12 - 40, 150 - 224.

[7] J. M. Gutierrez and V. Lanchares, *Elementos de matemática discreta*, Primera Ed. Servicio de Publicaciones Universidad de Rioja, 2010, pp. 12 - 15.

[8] S. Pérez, “Ecuaciones Diofanticas,” Universidad de Panamá, Ciudad de panamá, 2011.

[9] K. J. Hunt, “Polynomial optimization of discrete-time multivariable regulators,” *IEEE Proceedings-D*, vol. 139, no. 2, 1992.

[10] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, “A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems,” *Science Direct Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299-308, Jan. 2008.

[11] S. Grossman, “Ecuaciones Diofánticas,” in *Algebra lineal con aplicaciones*, Cuarta Edición, Ed. McGraw-Hill. 1996.

[12] T. Moir, D. Campbell, and H. Dabis, “A Polynomial Approach to Optimal and Adaptative Filtering with Application to Speech Enhancement,” *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1221-1224, 1991.

[13] R. Prokop, M. Dlapa, and R. Matusu, “Robust Control of Temperature in Hot-Air Tunnel,” in *16th Mediterranean Conference on Control and Automation Congress Centre*, 2008, pp. 576-581.

[14] W. A. Wolovich and P. J. Antsaklis, “The Canonical Diophantine Equiations With Applications,” *Society for Industrial and Applied Mathematics*, vol. 22, no. 5, pp. 777-787, 1984.

[15] E. Toro, Y. Restrepo, and M. Granada, “Adaptación de la técnica de particle swarm al problema de secuenciamiento de tareas,” in *Scientia et Technica Año XII No. 32*, 2006, no. 32, pp. 307-312.

[16] P. Dostil, V. Bobil, and M. Tomastik, “Application of Polynomial Method in Control of Time Delay Sistems,” *IEEE International Symposium on Computer Aided Control System Desing*, vol. September, 2004.

[17] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, “Hybrid metaheuristics in combinatorial optimization: A survey,” *Applied Soft Computing*, vol. 11, no. 6, pp. 4135-4151, Sep. 2011.

[18] A. Grosslinger and S. Schuster, “On Computing Solutions of Linear Diophantine Equations with One Non-linear Parameter,” in *2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2008, pp. 69-76.

[19] J. B. Peamn, C. T. Chen, and J. Murphy, “Algebraic Approach to Discrete Linear Control,” *IEEE Transactions on Automatic Control*, no. FEBRUARY, pp. 116 - 120, 1975.

[20] S. SHR Hua Wu, “Time - Varying Feedback Systems Design Via Diophantine Equation Order Reduction,” The University Of Texas at Arlington, 2007.

[21] J. Tse, J. Bentsman, and N. Miller, “Minimax Predictive Control,” in *Proceodings of the 31st Conference on Decision and Control*, 1992, no. December, pp. 65-70.

[22] M. E. Pérez and F. Herrera, “Algoritmos genéticos multimodales: Un estudio sobre la parametrización del método clearing aplicado al problema ‘Job shop’ ” Chile, 2003.

[23] S. Kemmoé Tchomté and M. Gourgand, “Particle swarm optimization: A study of particle displacement for solving continuous and combinatorial optimization problems,” *International Journal of Production Economics*, vol. 121, no. 1, pp. 57-67, Sep. 2009.

[24] I. Amaya, J. Cruz, and R. Correa, “Real Roots of Nonlinear Systems of Equations Through a Metaheuristic Algorithm,” *Revista Dyna*, vol. 78, no. 170, pp. 15-23, 2011.

[25] R. Parra Machío, “Ecuaciones Diofánticas,” in *Ecuaciones Diofánticas*, 2008, pp. 1-30.

[26] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization,” *Natural Computing*, vol. 8, no. 2, pp. 239-287, Sep. 2009.

[27] F. J. García, “Un Pequeño Manual para la Resolución de Problemas.” España, pp. 7 - 8, 2002.

[28] F. J. González, “Ecuaciones Diofánticas,” in *Apuntes de Matemática Discreta*, E. S. I.-D. De Matemáticas, Ed. Chile: Universidad de Cádiz, 2004, pp. 353 - 354.

# ANEXOS

## ANEXO A: DIAGRAMA DE FLUJO PARA PROBLEMAS CON MULTIPLES SOLUCIONES

Para problemas con múltiples soluciones, se realizó un ajuste al algoritmo del capítulo 4, se utiliza la nomenclatura de la tabla 1, y se presenta el diagrama de flujo en las figuras 23, 24, 25. A continuación una breve explicación de la figura 23.

- Los globos **A, B, C, D** permiten unir los diagramas de flujo de las figuras A1, A2 y A3 en un solo diagrama de flujo, con el fin de seguir la secuencia de diseño.
- Se introduce en la ejecución el número de partículas y el número máximo de iteraciones.
- Se definen en el algoritmo el número de incógnitas del problema, en este ejemplo son **x, y, z** es decir **3**.
- Se inicializa **x** y **v** como dos matrices de **np x nd** de números aleatorios enteros.
- Se introduce la función objetivo **fx** teniendo presente que **x1** es la primera variable en este caso **x =x1**, **xn** es la última variable en este caso **z = x3.**
- En la primera iteración **loc = x** y **floc = fx.**
- Se definen constantes de PSO Combinatorio **w = 0.75**, **c1 = 0.8**, **c2 = 0.2**, **alfa = 0.3.**
- Se inicializan valores necesitados **final = 0**, **final 2 = 0** y **k = 0.**
- Se encuentra **var1** como el **mínimo** de **fx** en todas las partículas, en Matlab se hace a través de la función min.
- Se iguala **var2** a la posición en la que encontramos este mínimo.
- Se define el global como el **x** en la posición del mínimo y el **fglo** como el **fx** en **var1**.

- Se define el número de soluciones que posee el problema y se reemplaza **S** por este número en el ciclo mientras que “do while” **cont<=S.**
- Se inicializan **final** y **final2** en la posición **cont** en **0,** es decir, **final(cont)=0** y **final2(cont)=0.**
- Se define la resolución del algoritmo, en este ejemplo es del **99%**, el ciclo mientras que “do while” **final<=0.99np.**
- Se definen **r1** y **r2** como números aleatorios de **0** a **1** y se acumula **k** con **k = k+1.**
- Se imprime la solución en la última iteración, la duración de la ejecución y la iteración en la que encontró la solución, cuando el **cont** sea mayor a **S**.
- Se pregunta si el **x** actual es igual al óptimo local y global simultáneamente, si es correcto, se iguala **y** actual o en **i,j,k** a **1** o **-1** aleatoriamente, si no es correcto, se pregunta si **x** es igual al óptimo global, si es correcto se iguala **y** a **1**, en caso contrario, se pregunta si es igual al óptimo local, si es correcto se iguala **y** a **-1**,si es incorrecto se iguala **y** a **0**.

En la figura 24 se tiene la continuación del diagrama de flujo del apéndice A, para entender el mismo, es necesario seguir el procedimiento:

- El globo **C** es la conexión entre la figura A1 y A2.
- Se inician dos ciclos para “for” uno de **i=1** hasta **np** acumulando **i** y otro de **j=1** hasta **nd** acumulando **j**.
- Se encuentra la velocidad **v** y el parámetro **B** “dummy” en la nueva iteración, a través de las formulas [10].
- Se inician dos ciclos para “for”, uno de **i=1** hasta **np** acumulando **i** y otro de **j=1** hasta **nd** acumulando **j**.
- Se pregunta si el parámetro “dummy” **B** es mayor que **alfa**, en caso correcto, se dice que el **x** en la nueva iteración es igual al óptimo global de la iteración actual; en caso incorrecto, se pregunta si el parámetro “dummy” **B** es menor

que **alfa**, en caso correcto, se asume **x** de la nueva iteración como el óptimo local de la iteración actual; en caso incorrecto se dice que **x** es un número aleatorio entero del espacio de búsqueda.

- Se inicia un ciclo para "for" de **i=1** hasta **np** acumulando **i**.
- Se escribe la función **fx** evaluada en la nueva iteración, es decir en k+1; para esto se tiene en cuenta, que **i** contiene la información del número de partículas; por tal, la primera variable, en este caso **x** es un vector de la forma **x(i,1,k+1)** . La última variable es un vector de la forma **x(i,n,k+1)** con **n** variables, en este ejemplo la variable es **z** y es de la forma **x(i,3,k+1)**.
- Se inician dos ciclos para "for", uno de **i=1** hasta **np** acumulando **i** y otro de **j=1** hasta **nd** acumulando **j**.
- Se pregunta si **fx** en la nueva iteración, es menor que el **floc** en la iteración actual, en caso correcto, se dice que el nuevo óptimo local es igual al **x** en la nueva iteración; en caso incorrecto, se dice que el nuevo óptimo local es igual al anterior óptimo local.

En la figura 25 se tiene la continuación del diagrama de flujo del apéndice A, para entender el mismo, es necesario seguir el siguiente procedimiento:

- El globo **D** es la conexión entre la figura A2 y A3.
- Se inicia un ciclo para "for" de **i=1** hasta **np** acumulando **i**. Se escribe la función **floc**, que corresponde a los locales evaluados en el **fx** de la nueva iteración, es decir en **k+1**, se tiene en cuenta que **i** contiene la información del número de partículas. La primera variable en este caso **x**, es un vector de la forma **loc(i,1,k+1)**, la última variable es un vector de la forma **loc(i,n,k+1)**, con **n** variables, en este caso **z** y es de la forma **loc(i,3,k+1)**.
- Se encuentra **var1** y **var2**.

- Se define el óptimo global como el óptimo local en la posición **var2**. Se iguala **fglo** a **var1**.
- Se pregunta si **final=1** y **final2=0**, en caso correcto, se dice que **final2=k**, en caso incorrecto, se pregunta si **k=ni**, en caso correcto se dice que **final=np**, en caso incorrecto, se pregunta si **cont=1**, en caso correcto se iguala **final** a **np,** en caso incorrecto se acumula **i** hasta **cont-1** en un ciclo para o "for". Si esta dentro del ciclo se pregunta si **soluc** es igual a **solu,** en caso correcto, se hace **comp=comp+1**, en caso incorrecto se sale del ciclo.
- Se pregunta si **comp=0**, si es correcto se hace **soluc=solu** y **cont=cont+1**, si es falso ir a globo **B**.

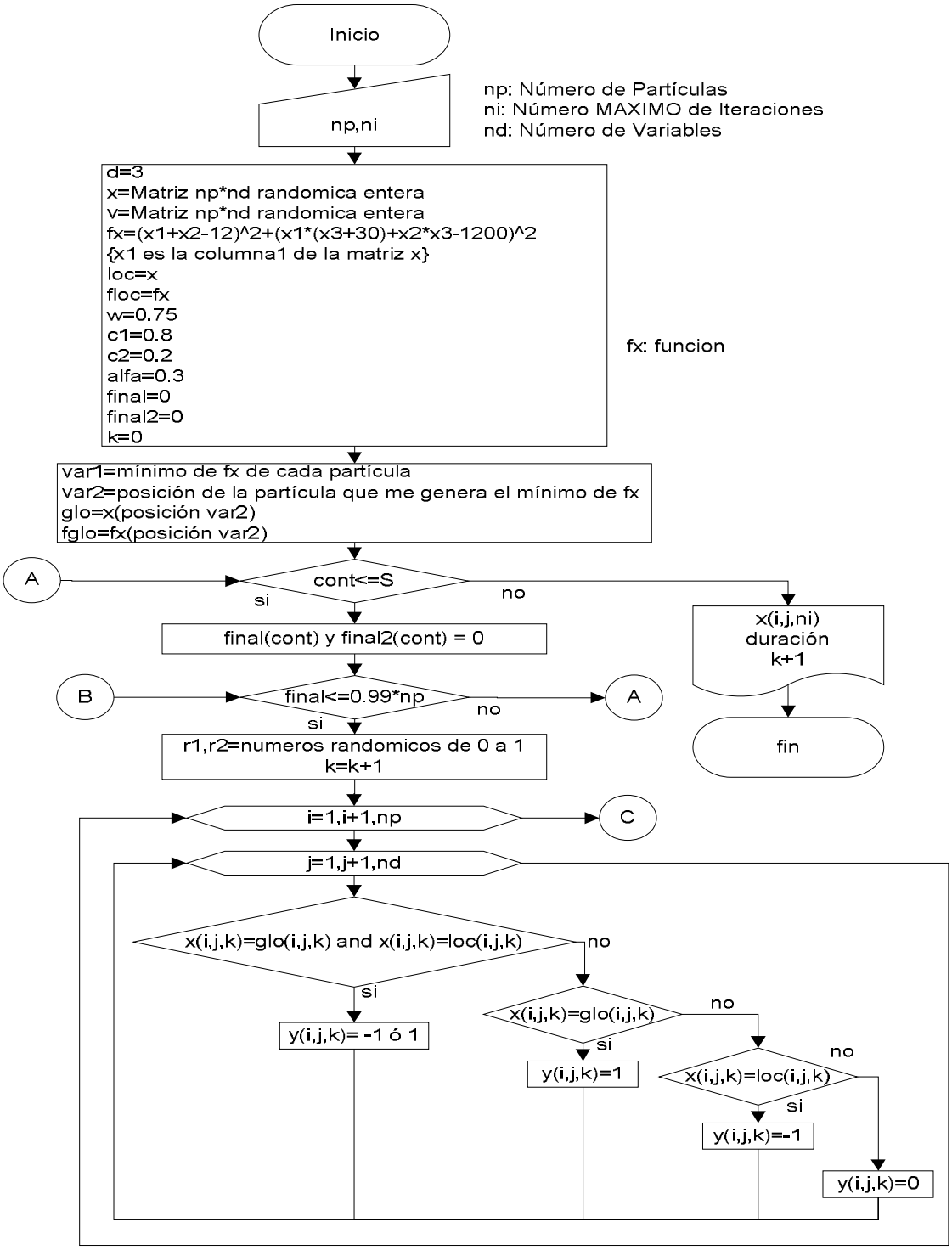


**Figura 23 Primera parte del diagrama de flujo del algoritmo con múltiples soluciones**

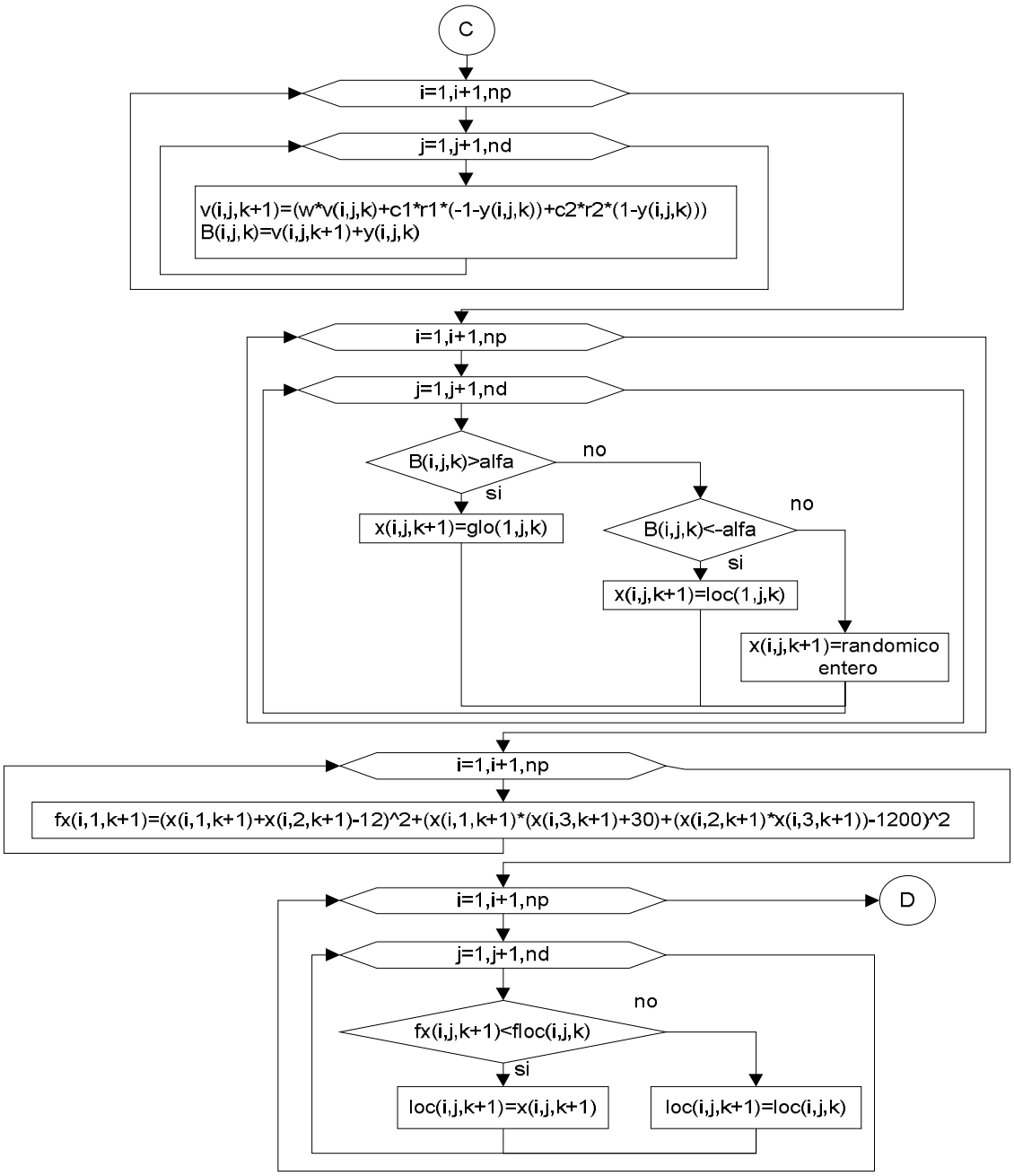


**Figura 24. Segunda parte del diagrama de flujo del algoritmo con múltiples soluciones**

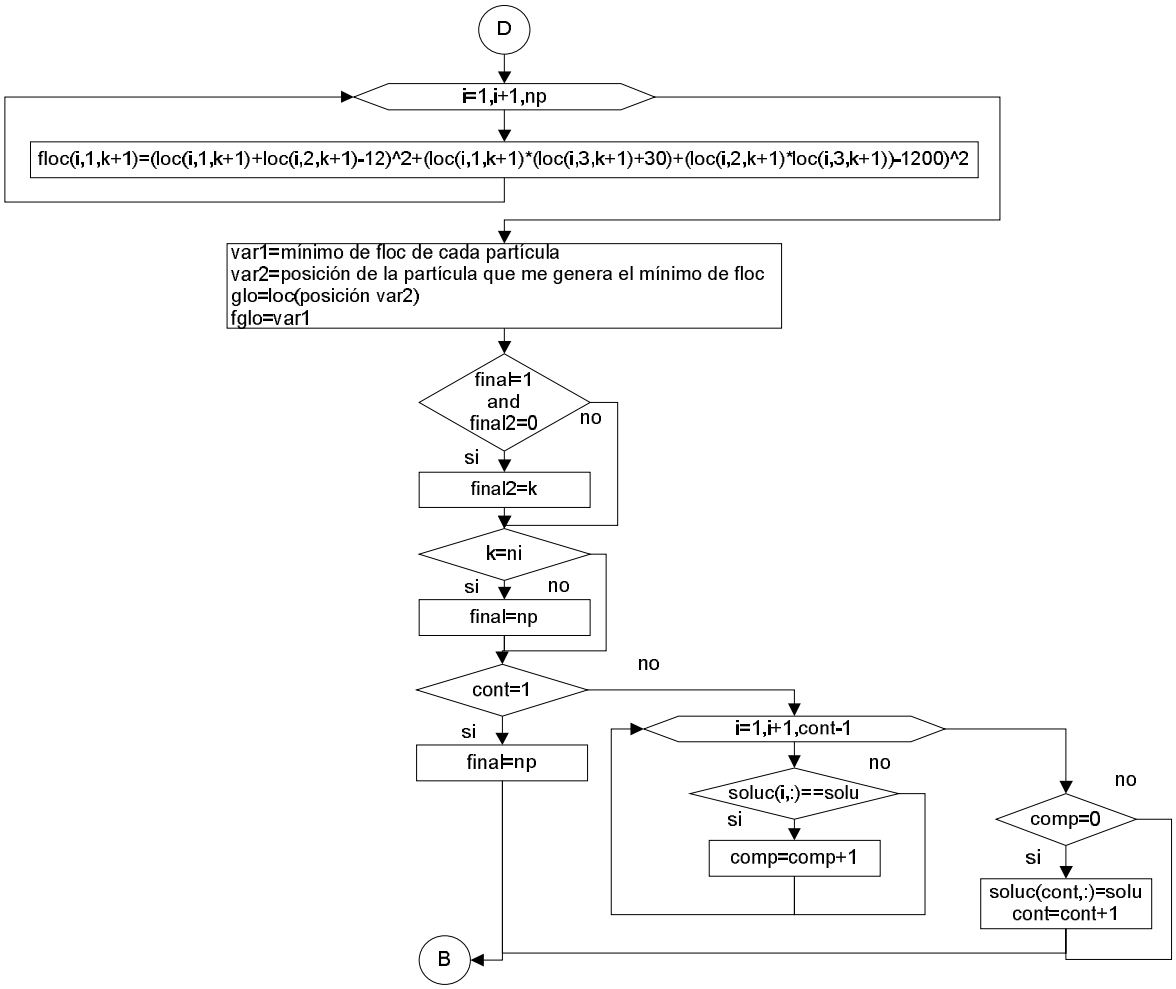


**Figura 25. Tercera parte del diagrama de flujo del algoritmo con múltiples soluciones**