

Ingeniería Técnica Industrial Mecánica

METODOLOGÍA PARA LA ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES A UN CONJUNTO DE VARIABLES DEPENDIENTES



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INGENIERÍA MECÁNICA

Ingeniería Técnica Industrial: Mecánica

Julio 2010

Autor: Jesús González García.

Tutora: Isabel Marina González Farias.

ÍNDICE

1. CAPITULO PRIMERO: INTRODUCCION	3
1.1. JUSTIFICACIÓN Y ALCANCE DEL PROYECTO.....	3
1.2. OBJETIVOS DEL PROYECTO.....	4
1.3. ESTRUCTURA DEL PROYECTO.....	4
2. CAPITULO SEGUNDO: ESTADO DEL ARTE	5
2.1. CONCEPTOS ESTADÍSTICOS IMPORTANTES.....	5
2.1.1. <i>Media</i>	5
2.1.2. <i>Desviación estándar</i>	5
2.1.3. <i>Varianza</i>	5
2.1.4. <i>Covarianza</i>	6
2.1.5. <i>Índice de capacidad</i>	6
2.1.6. <i>Matriz de covarianzas</i>	6
2.1.7. <i>Análisis de datos multivariantes</i>	7
2.1.8. <i>Distribución normal univariante y multivariante</i>	7
2.1.9. <i>Intervalos de confianza univariantes bajo normalidad</i>	10
2.1.10. <i>Región de confianza multivariante bajo normalidad</i>	11
2.1.11. <i>Obtención de intervalos de confianza independientes a partir de una región de confianza multivariante bajo normalidad</i>	13
2.1.12. <i>Análisis de componentes principales</i>	15
2.2. INTRODUCCIÓN AL DISEÑO O ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES.....	16
2.2.1. <i>Definición de tolerancias de fabricación</i>	16
2.2.2. <i>Definición del problema de asignación estadística óptima de tolerancias dimensionales y valores nominales</i>	17
2.3. ASIGNACIÓN ESTADÍSTICA DE TOLERANCIAS CONSIDERANDO VARIABLES INDEPENDIENTES. MÉTODOS ACTUALES.....	20
2.3.1. <i>Metodología utilizada bajo independencia</i>	20
2.4. ASIGNACIÓN ESTADÍSTICA DE VALORES NOMINALES CONSIDERANDO VARIABLES INDEPENDIENTES. METODOS ACTUALES.....	23
2.4.1. <i>Metodología utilizada bajo independencia</i>	23
2.5. ASIGNACIÓN ESTADÍSTICA DE TOLERANCIAS CONSIDERANDO VARIABLES DEPENDIENTES.METODOS ACTUALES.....	24

3. CAPÍTULO TERCERO: ALGORITMOS GENÉTICOS Y SU USO EN MATLAB.....	25
3.1. INTRODUCCION A LOS ALGORITMOS GENETICOS (AG).....	25
3.1.1. Breve reseña histórica.....	26
3.1.2. Descripción del algoritmo de optimización.....	27
3.1.3. Estructura de los algoritmos genéticos.....	28
3.1.3.1. Codificación.....	29
3.1.3.2. Creación de la población inicial.....	31
3.1.3.3. Función de evaluación y función de aptitud.....	31
3.1.3.4. Selección.....	32
3.1.3.5. Cruce.....	32
3.1.3.6. Mutación.....	33
3.1.3.7. Reemplazo.....	34
3.1.3.8. Criterio de parada.....	35
3.1.4. Ejemplo de implementación de un algoritmo genético.....	35
3.2. ALGORITMOS GENÉTICOS CON MATLAB.....	40
3.2.1. Toolbox de algoritmos genéticos en MATLAB.....	40
3.2.2. Ejemplo de implementación.....	43
3.2.3. Ejemplo práctico de aplicación.....	47
3.2.3.1. Descripción del problema.....	47
3.2.3.2. Asignación de las variables y límites.....	48
3.2.3.3. Planteamiento del problema para la aplicación de AG.....	50
3.2.3.4. Construcción de la función objetivo.....	51
3.2.3.5. Programación en MATLAB.....	51
3.2.3.6. Parámetros de AGs usados.....	52
3.2.3.7. Conclusión.....	53
4. CAPÍTULO CUARTO: METODOLOGÍA PARA LA ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES A UN CONJUNTO DE VARIABLES DEPENDIENTES.....	54
4.1. DEFINICIÓN DEL PROBLEMA.....	54
4.2. METODOLOGIA PROPUESTA PARA LA ASIGNACION DE TOLERANCIAS Y VALORES NOMINALES.....	57
4.2.1. Etapa I.....	58
4.2.2. Etapa II.....	64
5. CAPÍTULO QUINTO: EJEMPLOS DE APLICACIÓN DE LA METODOLOGÍA.....	68
5.1. ASIGNACION DE TOLERANCIAS Y VALORES NOMINALES EN UNA PIEZA: DEPÓSITO CILÍNDRICO.....	68
5.2. ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES EN UN ENSAMBLE DE BLOQUES.....	79
5.3 ASIGNACION DE TOLERANCIAS Y VALORES NOMINALES EN UN ENSAMBLE DE CILINDROS.....	88
6. CAPITULO SEXTO: CONCLUSIONES Y FUTURAS LINEAS DE TRABAJO.....	95
6.1. CONCLUSIONES.....	95
6.2. FUTURAS LÍNEAS DE TRABAJO.....	96
BIBLIOGRAFIA.....	97
ANEXOS.....	100

1. CAPÍTULO PRIMERO: INTRODUCCIÓN

1.1. JUSTIFICACIÓN Y ALCANCE DEL PROYECTO.

La asignación de tolerancias ha evolucionado hasta convertirse en un factor de vital importancia dentro de la fabricación mecánica. En la actualidad esta asignación depende de varios factores entre los que se puede destacar el proceso de fabricación empleado.

A partir de la introducción de la estadística en el control de los procesos de fabricación, se comenzó a relacionar la variabilidad de dichos procesos con la asignación de tolerancias con el objetivo de adecuar la fabricación de una pieza a un determinado proceso, mejorando considerablemente el ratio coste-calidad en la asignación de tolerancias.

Las tolerancias pueden ser diferenciadas en dos grupos, las tolerancias geométricas que hacen referencia a las cualidades externas, centrándose, como su propio nombre indica en las desviaciones geométricas de la pieza respecto de las que idealmente se esperan y las tolerancias dimensionales que hacen referencia a los límites dados para los valores de las distintas dimensiones de las que se compone una pieza o conjunto. Ambos tipos de tolerancias son necesarias para la correcta funcionalidad de las piezas o conjuntos mecánicos y sus valores dependerán de los procesos productivos y de los requerimientos de calidad que se establezcan previamente.

La asignación de tolerancias está estrechamente relacionada con el proceso de fabricación y el control de calidad del producto. La asignación de unas tolerancias muy pequeñas repercutirá directamente en la necesidad de trabajar con procesos productivos costosos, de baja variabilidad que aumentan de manera considerable los costes de producción. Por el contrario, la asignación de tolerancias muy amplias producirá el aumento de elementos que no cumplen con las especificaciones establecidas, aumentando la proporción de defectuosos y repercutiendo en los costes de calidad.

En este proyecto nos centraremos en la asignación óptima de tolerancias dimensionales. Se considera que unas tolerancias son óptimas cuando son las máximas tolerancias posibles que permiten cumplir las restricciones de la pieza. Para el cumplimiento de las restricciones de la pieza, se considerará el enfoque estadístico. Es decir, no se espera que todas las piezas cumplan las restricciones, si no que se tenga una proporción de defectuosos muy pequeña.

Los métodos tradicionales para la asignación de tolerancias dimensionales están basados en la independencia entre variables. Además, consideran que las tolerancias son simétricas con respecto a un valor nominal que es fijo. En la asignación de las tolerancias estos métodos usan la variabilidad del proceso de fabricación, expresada como varianza de las variables, sin embargo esta variabilidad es ficticia ya que no se basa en datos tomados del proceso de fabricación. Un método innovador para la asignación de tolerancias dimensionales óptimas considerando correlación entre variables correladas (debido al proceso de fabricación) ha sido propuesto por González y Sánchez (2008). Este método se basa en obtener unas variables independientes a partir de las variables correladas originales, y resolver el problema de asignación óptima de las tolerancias usando estas variables independientes. El método hace uso de la matriz de covarianzas del proceso de fabricación que debe ser estimada con datos de dicho proceso. Al igual que en los métodos tradicionales en este caso se considera de los límites de tolerancias son simétricos respecto a un valor nominal que es fijo.

En este proyecto, se propondrá una metodología basada en la propuesta de González y Sánchez (2008). El procedimiento para la asignación óptima de tolerancias considerará correlación entre variables, pero incluirá la posibilidad de asignar valores nominales de las variables, dentro de un rango fijo, de forma que se maximicen las tolerancias.

Debido al elevado número de variables a optimizar que se pueden presentar a la hora de resolver esta metodología, se planteará su resolución mediante la aplicación de algoritmos genéticos. Este método de optimización basado en las técnicas evolutivas postuladas por Charles Darwin en 1859, permite solucionar problemas complejos de optimización de forma rápida y sencilla sin la necesidad de tener conocimientos específicos previos del problema. La aplicación de algoritmos genéticos como algoritmo de optimización se llevará a cabo mediante la toolbox Gatool de MATLAB.

1.2. OBJETIVOS DEL PROYECTO.

El objetivo principal de este proyecto es presentar de manera detallada pero sencilla, una nueva metodología para la asignación óptima de valores nominales y tolerancias a un conjunto de variables correladas. Esta asignación óptima busca maximizar las tolerancias de manera que se obtenga una proporción de piezas defectuosas pequeña.

Se pretende que cualquier persona, incluso aquéllas que no poseen ningún conocimiento sobre este tema pueda comprender las bases de la metodología propuesta así como aplicarla a problemas reales.

Se realizará una descripción de la aplicación de Algoritmos Genéticos como método de optimización así como una guía rápida de utilización dentro del programa matemático de cálculo computacional MATLAB, que permita a cualquier persona, aunque no haya tenido contacto con este método de optimización, la resolución de problemas sencillos.

Por último se mostrarán varios ejemplos de aplicación para una mayor comprensión de la metodología y se darán a conocer las ventajas y desventajas de la aplicación de esta nueva metodología con respecto a las diferentes metodologías propuestas con anterioridad y las futuras líneas de investigación que se podrían plantear para mejorarla.

1.3. ESTRUCTURA DEL PROYECTO.

En el capítulo II, se describen los conceptos más importantes relacionados con la asignación de tolerancias y valores nominales, así como, las metodologías actuales utilizadas para la asignación de éstas.

En el capítulo III, se describe brevemente la técnica de optimización mediante algoritmos genéticos, que será usada para la resolución del problema de asignación de tolerancias y valores nominales. También se presenta el uso de algoritmos genéticos en Matlab y un ejemplo práctico de optimización.

En el capítulo IV se desarrolla la metodología propuesta de forma detallada, definiendo las etapas de las que se compone paso por paso.

En el capítulo V, se muestra la aplicación completa de la metodología y los resultados obtenidos sobre tres ejemplos prácticos.

En el capítulo VI, se exponen las conclusiones y las futuras líneas de trabajo que se podrían emplear para la mejora de la metodología.

Por último, se incluye la bibliografía y los enlaces web consultados así como una serie de anexos con información útil para el lector.

2. CAPÍTULO SEGUNDO: ESTADO DEL ARTE

En este capítulo se tratarán conceptos teóricos importantes relacionados con la metodología que será propuesta en este proyecto. El capítulo está estructurado en los siguientes apartados. El primero incluye la definición de una serie de conceptos estadísticos importantes. En el segundo apartado se define el problema de la asignación de tolerancias y valores nominales, objeto de este proyecto. Por último, en el tercero, se presentan brevemente los procedimientos que existen actualmente para resolver este problema.

2.1. CONCEPTOS ESTADÍSTICOS IMPORTANTES.

2.1.1. Media.

Dado un conjunto de n datos numéricos x_1, x_2, \dots, x_n , se define la media aritmética como:

$$\bar{x} = \frac{\sum x_i}{n} \quad (2.1)$$

Donde el sumatorio se extiende al número total de datos n .

2.1.2. Desviación estándar.

La desviación estándar o desviación típica es una medida de centralización o dispersión. Es de gran utilidad en la estadística descriptiva. Junto a la varianza con la que está estrechamente relacionada, es una medida (cuadrática) que informa de la media de distancias que tienen los datos respecto de su media aritmética, expresada en las mismas unidades que la variable.

Para conocer con detalle un conjunto de datos, no basta con conocer las medidas de tendencia central, sino que necesitamos conocer también la desviación que representan los datos en su distribución respecto de la media aritmética de dicha distribución, con objeto de tener una visión de los mismos más acorde con la realidad a la hora de describirlos e interpretarlos para la toma de decisiones.

La desviación típica muestral es igual a:

$$\sigma = \sqrt{\frac{\sum x_i - \bar{x}}{n}} \quad (2.2)$$

2.1.3. Varianza.

La varianza es una medida de la dispersión de una variable aleatoria respecto a su esperanza. Está relacionada con la desviación estándar o desviación típica, que se suele denotar por la letra griega σ (sigma) y que es la raíz cuadrada de la varianza.

La varianza es igual a:

$$\sigma^2 = \frac{\sum x_i - \bar{x}}{n} \quad (2.3)$$

2.1.4. Covarianza.

En el conjunto de n -variables continuas, la covarianza es la medida de la relación lineal entre cada par de variables, definida por:

$$Cov(x, y) = \frac{\sum x_i - \bar{x} \cdot y_i - \bar{y}}{n} , \quad (2.4)$$

Donde el sumatorio está extendido a las n parejas de valores x, y .

2.1.5. Índice de capacidad.

El índice de capacidad del proceso, denotado como C_p es un cálculo estadístico sobre la capacidad de un proceso (proporción de defectuosos) para producir una pieza dentro de unos límites predefinidos (TS , tolerancia superior y TI , tolerancia inferior). Este índice juega un papel fundamental en las plantas de producción a la hora de demostrar que un proceso (ej. de producción de tornillos) es fiable y está bajo control. Se define como:

$$C_p = \frac{TS - TI}{6\sigma} \quad (2.5)$$

Bajo normalidad, y si el proceso está centrado en la media, un valor del índice de capacidad igual a la unidad, $C_p = 1$, indica que la proporción de defectuosos esperada es igual a $p = 0.0027$. Como regla general, un índice mayor a la unidad, indica que el proceso es capaz $p < 0.0027$. Un índice menor a la unidad, indica que el proceso no es capaz $p > 0.0027$.

2.1.6. Matriz de covarianzas.

En estadística y teoría de la probabilidad, la matriz de covarianza es una medida multivariante compuesta por una matriz cuadrada simétrica que tiene en la diagonal principal las varianzas de las observaciones, y fuera de ellas las covarianzas entre variables.

La matriz de covarianza es una herramienta muy útil en varios campos. A partir de ella se puede derivar una transformación lineal que puede de-correlacionar los datos o, desde otro punto de vista, encontrar una base óptima para representar los datos de forma óptima, esto es denominado como análisis de componentes principales (ACP).

$$\Sigma = \begin{bmatrix} \sigma_1^2 & Cov_{12} & \cdots & Cov_{1k} \\ Cov_{21} & \sigma_2^2 & \cdots & Cov_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ Cov_{k1} & Cov_{k2} & \cdots & \sigma_k^2 \end{bmatrix} \quad (2.6)$$

2.1.7. Análisis de datos multivariantes.

El análisis de datos multivariantes comprende el estudio estadístico de varias variables medidas en elementos de una población. Algunos de los objetivos de un análisis multivariante pueden ser, por ejemplo, los siguientes:

1. Resumir los datos mediante un pequeño conjunto de nuevas variables, construidas como transformaciones de las originales, con la mínima pérdida de información, que nos permitirán:
 - Comparar distintos conjuntos de datos mediante su representación gráfica.
 - Mejorar nuestro conocimiento de la realidad estudiada.
2. Identificar posibles grupos que puedan existir dentro de las observaciones. En muchas situaciones los grupos son desconocidos a priori y queremos disponer de un procedimiento objetivo para obtenerlos.
3. Clasificar nuevas observaciones dentro de los grupos definidos a priori.
4. Estudiar la relación entre dos conjuntos de variables, ya sean iguales medidas en dos momentos diferentes o distintos, y conocer cuantas dimensiones tiene esta relación.

El análisis multivariante de datos proporciona métodos objetivos para conocer cuantas variables indicadoras o factores, son necesarias para describir adecuadamente una realidad compleja y determinar las propiedades de estas variables.

2.1.8. Distribución normal univariante y multivariante.

Cuando se trabaja en la vida real, la suposición más habitual es que una variable sigue una distribución normal o gaussiana univariante. La justificación matemática de esta distribución se encuentra en el Teorema Central del Límite que demuestra que la suma de variables independientes se distribuye en el límite como una normal. Muchas características que se miden son la conjunción de muchas causas que actúan conjuntamente sobre el suceso. Por ejemplo, la altura de las personas se considera que se distribuye como una normal ya que su valor es debido a múltiples causas ambientales, alimentarias y genéticas.

La distribución normal univariante es el modelo de distribución de probabilidad para variables continuas más importante y está definido por su función de densidad.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2} (x - \mu)^2\right] \quad (2.7)$$

y su representación gráfica se puede ver en la figura 2.1.

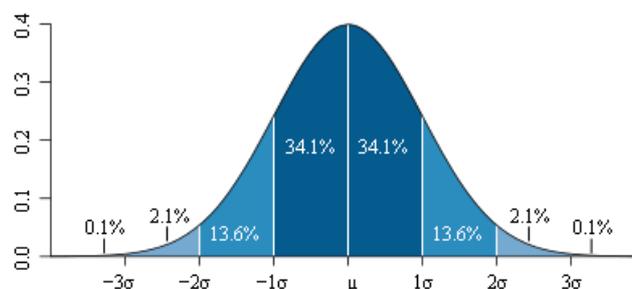


Figura 2.1: Distribución normal univariante.

La distribución normal multivariante, también llamada distribución gaussiana multivariante, es una generalización de la distribución normal univariante a dimensiones superiores.

En este caso, un conjunto de k variables $\mathbf{X} = X_1, X_2, \dots, X_k$, sigue una distribución normal k -dimensional, $\mathbf{X} \approx N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, si su función de densidad conjunta obedece a la expresión.

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_k) = \frac{e^{-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}]' \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}}{\sqrt{2\pi^k \det \boldsymbol{\Sigma}}} \quad (2.8)$$

Donde, $\boldsymbol{\mu}$ es el vector de medias $\boldsymbol{\mu} = \mu_1, \mu_2, \dots, \mu_k$ y $\boldsymbol{\Sigma}$ es la matriz de covarianzas definida por la matriz (2.6).

Bajo la hipótesis de normalidad multivariante, se tiene que las observaciones de las variables definen un hiperelipsoide de dimensión k .

Por ejemplo, para el caso de $k=2$, la región definida es una elipse como la que se representa en la figura 2.2.

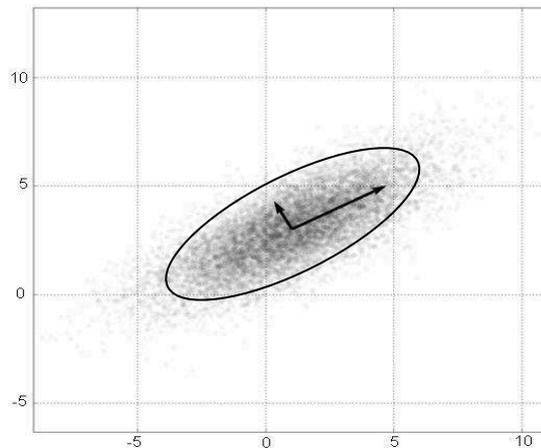


Figura 2.2: Representación de la dispersión de los datos bajo normalidad multivariante.

Propiedades de la distribución normal multivariante.

1. Si un conjunto de variables sigue una distribución normal multivariante, puede demostrarse que todas las distribuciones marginales son normales, de forma que cada variable es una normal $X_i \approx N(\mu_i, \sigma_i^2)$. Igualmente el resultado recíproco se cumple también: dadas k variables aleatorias normales, su distribución conjunta es una normal k -dimensional.
2. Puede demostrarse que las distribuciones condicionadas de cualquier dimensión son también normales y que, en particular, las de dimensión uno son normales unidimensionales que tienen por media el valor esperado por la regresión lineal, y por varianza la varianza residual, de esa regresión.
3. Un caso importante de distribución normal k -dimensional es aquél, en el que todas las variables son incorreladas. En este caso todas las covarianzas, serán nulas y la matriz de covarianzas será diagonal.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k^2 \end{bmatrix} \quad (2.9)$$

4. Un resultado aún más importante es que bajo normalidad, la no correlación (covarianzas iguales a 0) implica independencia de las variables. Esto no es necesariamente cierto si las variables no siguen una distribución normal.
5. Dadas k variables $\mathbf{X} = X_1, X_2, \dots, X_k$, distribuidas como una normal k -dimensional, $\mathbf{X} \approx N_k(\boldsymbol{\mu}, \Sigma)$, y dada la matriz de transformación \mathbf{A} , definida por:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{k1} \\ a_{12} & a_{22} & \cdots & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{km} \end{bmatrix} \quad (2.10)$$

el nuevo conjunto de variables m -dimensional, $\mathbf{Y} = \mathbf{A}\mathbf{X}$ seguirá una distribución:

$$\mathbf{Y} \approx N_m(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\Sigma\mathbf{A}') \quad (2.11)$$

2.1.9. Intervalos de confianza univariantes bajo normalidad.

Se llama intervalo de confianza a un par de números entre los cuales se estima que estará cierto valor desconocido con una determinada probabilidad de acierto. Formalmente, estos números determinan un intervalo que se calcula a partir de datos de una muestra y el valor desconocido es un parámetro poblacional, por ejemplo la media.

La probabilidad de éxito en la estimación se representa por $1-\alpha$ y se denomina nivel de confianza. En estas circunstancias, α es llamada nivel de significación y representa la probabilidad de fallar en la estimación mediante tal intervalo.

El nivel de confianza y la amplitud del intervalo varían conjuntamente, de forma que un intervalo más amplio tendrá más posibilidades de acierto (mayor nivel de confianza), mientras que para un intervalo más pequeño, que ofrece una estimación más precisa, aumentan sus posibilidades de error.

Por ejemplo, el intervalo para la media se puede construir de la siguiente manera. Conocida la varianza de la población, sabemos que el error relativo de estimación de la media μ , mediante la media muestral \bar{x} es una variable normal estándar z .

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \quad (2.12)$$

Esta variable z , permite construir el intervalo de confianza:

$$P\left(-z_{\alpha/2} \leq \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \leq z_{\alpha/2}\right) = 1 - \alpha$$

Donde $z_{\alpha/2}$, es un valor normal estándar tal que:

$$P(z > z_{\alpha/2}) = 1 - \Phi(z_{\alpha/2}) = \alpha/2$$

Siendo Φ , la función de distribución normal estándar. Entonces el intervalo será:

$$\bar{x} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

que tendrá por confianza $1-\alpha$. Los valores se han escogido simétricos para que conduzcan al intervalo más corto posible.

La distribución de confianza en este caso resulta al despejar μ , en la ecuación (2.12) para obtener:

$$\mu = \bar{x} + z \frac{\sigma}{\sqrt{n}} \quad (2.13)$$

Al variar α , como \bar{x} se supone constante, la única variable aleatoria es z ; por lo tanto, la distribución generada es normal, con media \bar{x} y varianza σ/\sqrt{n} . Esta distribución resume la incertidumbre existente respecto al valor desconocido μ . Ver figura 2.3.

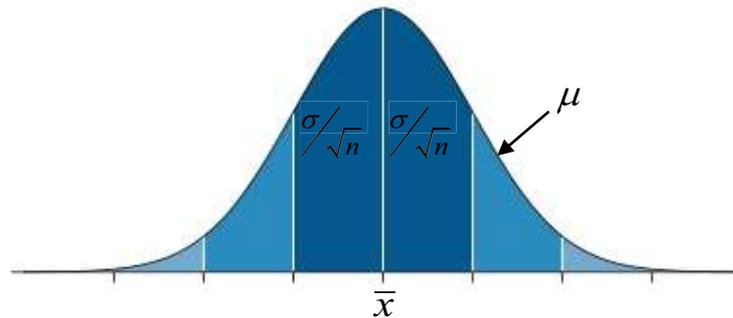


Figura 2.3: Distribución de intervalos univariantes bajo normalidad.

2.1.10. Región de confianza multivariante bajo normalidad.

Es la generalización de intervalo de confianza al espacio multi-dimensional, es decir, al análisis conjunto de varias variables.

Si manejamos dos variables independientes, analizadas de forma independiente, la región de confianza tendría la forma de la figura 2.4.

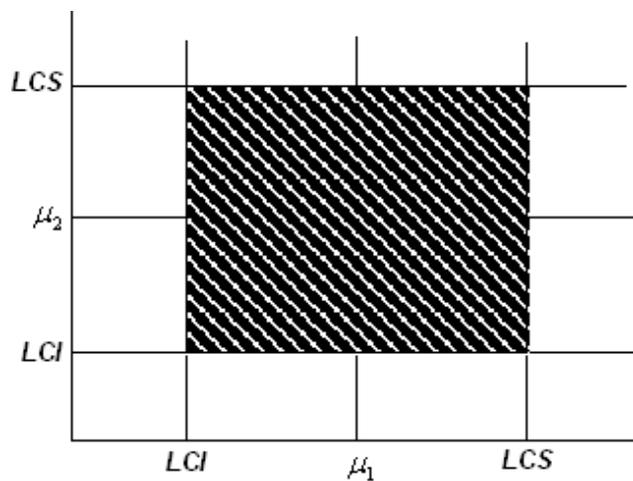


Figura 2.4: Región de confianza para dos variables independientes analizadas individualmente.

Si la representación de la región de confianza se realiza de forma conjunta (análisis multivariante) la región tendrá forma de elipse, cuya inclinación dependerá de la correlación existente entre las variables. Si las variables son independientes la región de confianza tomará la forma de la figura 2.5.

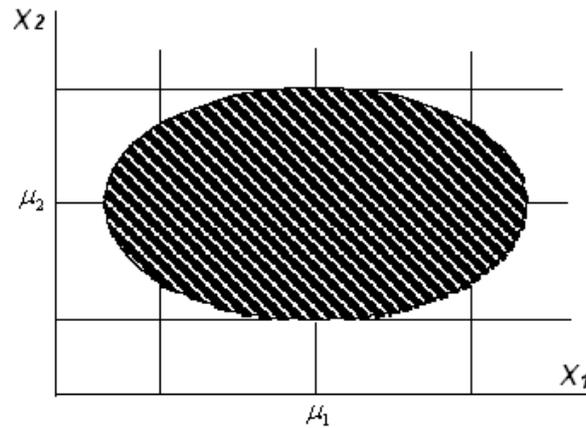


Figura 2.5: Región de confianza para dos variables independientes analizadas de forma conjunta.

Si, por el contrario las variables son dependientes, la forma de la región de confianza dependerá de la cuantía de la correlación y de su signo. Por ejemplo, para dos variables positivamente correlacionadas, la elipse que define la región de confianza se puede ver en la figura 2.6.

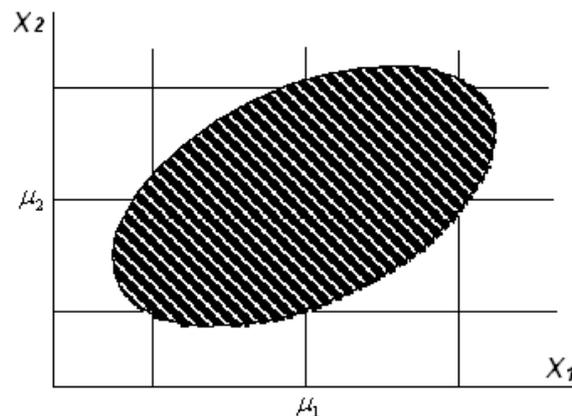


Figura 2.6: Región de confianza para dos variables dependientes analizadas de forma conjunta.

2.1.11. Obtención de intervalos de confianza independientes a partir de una región de confianza multivariante bajo normalidad.

Para obtener los intervalos de confianza independientes de cada una de las dimensiones X_1, X_2, \dots, X_k a partir de una región de confianza multivariante bajo normalidad, aplicaremos la definición de Nickerson (1994), basada en los trabajos de Working y Hotelling (1929) y Scheffe (1959), en la que, como se puede ver en la figura 2.7, se estudia la proyección del hiperelipsoide que contiene la región de confianza multivariante bajo la asunción de normalidad, correspondiente a un hiperrectángulo centrado en la media y tangente al hiperelipsoide.

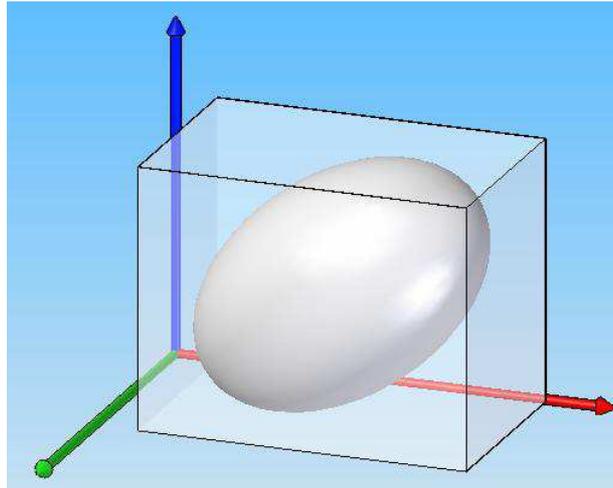


Figura 2.7: Proyección del hiperelipsoide en un hiperrectángulo según Nickerson (1994).

Por lo tanto, mediante la aplicación de la definición de Nickerson (1994), tenemos un hiperrectángulo, tangente al hiperelipsoide cuyas semi-longitudes de sus caras, l_j pueden ser obtenidas como:

$$l_j = \sqrt{\Sigma^{-1}_{jj} \chi^2_{m,1-\alpha}} \quad (2.14)$$

Donde el subíndice jj denota el j -ésimo elemento de la diagonal inversa de Σ y $\chi^2_{m,1-\alpha}$ es el valor correspondiente a la proporción deseada en la distribución Chi-cuadrado con k grados de libertad.

Las caras de este hiperrectángulo, están definidas por los límites $[LI_j, LS_j]$ y serán obtenidas como:

$$LI_j = \mu_j - l_j$$

$$LS_j = \mu_j + l_j$$

La figura 2.8 muestra un ejemplo de este procedimiento en dos dimensiones. La región **A** representa la elipse con la concentración de datos correspondiente y la región **B** representa el rectángulo obtenido de la proyección de la elipse utilizando Nickerson (1994).

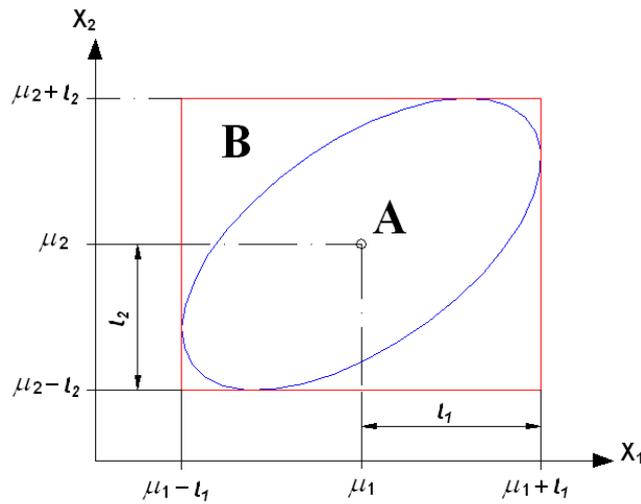


Figura 2.8: Modelo de proyección de Nickerson para $k = 2$.

Este hiperrectángulo definido por Nickerson será utilizado en la metodología para definir el hiperrectángulo de tolerancias y la aproximación a la proporción de no conformidad.

La región de tolerancias estará definida por los intervalos LTI_i, LTS_i correspondientes al hiperrectángulo de dimensión k , que será relacionado al hiperelipsoide definido por la matriz de covarianzas.

2.1.12. Análisis de componentes principales.

El análisis de componentes principales, desde ahora (ACP), es una técnica estadística de síntesis de la información que fue inicialmente desarrollada por Pearson a finales del siglo XIX y que posteriormente fue estudiada por Hotelling en los años 30 del siglo XX. Sin embargo hasta la aparición de los ordenadores, no se empezaron a popularizar.

Este análisis se utiliza principalmente para pasar de variables dependientes a independientes (bajo la hipótesis de normalidad multivariante) y también, para reducir la dimensión de un problema, pasando de un número mayor de variables a un número menor, perdiendo la menor cantidad de información.

Consideremos una serie de k variables $\mathbf{X} = X_1, X_2, \dots, X_k$, que definen ciertas magnitudes de un conjunto mecánico. Estas variables, pueden considerarse como una combinación lineal de k factores independientes $\mathbf{Z} = Z_1, Z_2, \dots, Z_k$ relacionados con el proceso de fabricación, conocidos dentro del análisis multivariante como factores latentes.

Para encontrar estos k factores independientes \mathbf{Z} , es posible aplicar ACP. Para ello, la matriz de covarianzas denominada por Σ_X , se descompone en valores singulares como:

$$\Sigma_X = \mathbf{C}\mathbf{D}\mathbf{C}' \quad (2.15)$$

Donde \mathbf{C} es la matriz de autovectores $k \times k$, y \mathbf{D} es la matriz diagonal de autovalores $\lambda_1, \lambda_2, \dots, \lambda_k$. Estos autovalores son las varianzas de los factores independientes \mathbf{Z} , los cuales son iguales a:

$$\mathbf{Z} = \mathbf{C}\mathbf{X} \quad (2.16)$$

Dentro del análisis multivariante, Los autovectores representan las direcciones de los ejes del hiperelipsoide definido por la matriz de covarianzas Σ_X y los autovalores representan el tamaño de estos ejes. Por tanto, un cambio en los autovalores producirá un hiperelipsoide más grande o más pequeño en cualquiera de sus direcciones, mientras que un cambio en los autovectores producirá un cambio en la dirección de los ejes.

2.2. INTRODUCCIÓN AL DISEÑO O ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES.

2.2.1. Definición de tolerancias de fabricación.

Según la Real Academia Española, la definición de tolerancia es:

“Máxima diferencia que se tolera o admite entre el valor nominal y el valor real o efectivo en las características físicas y químicas de un material, pieza o producto”.

A la hora de realizar una pieza dentro de un proceso de fabricación, observaremos necesariamente diferencias entre las dimensiones reales obtenidas tras su fabricación y el valor nominal que se había indicado inicialmente, si este valor real se encuentra comprendido entre unos límites (máximo y mínimo), establecido con anterioridad, la medición podrá considerarse como válida. A este error admitido se le denomina tolerancia y está estrechamente relacionado con las variables de coste del producto y calidad, de esta forma cuanto más se ajusten los límites de tolerancia, mayor será la calidad del producto y mayor será el coste de éste. Estas tolerancias permiten cumplir con el objetivo de intercambiabilidad y correcto montaje de piezas realizadas en serie.

Dentro de la definición de tolerancias podemos diferenciar dos tipos:

Las tolerancias geométricas, que hacen referencia a las cualidades externas de las piezas y, como su propio nombre indica, se centran en las desviaciones geométricas de la pieza respecto de las que idealmente se esperan y se pueden dividir en tolerancias de forma, orientación, posición y oscilación, como se puede ver en la figura 2.9.

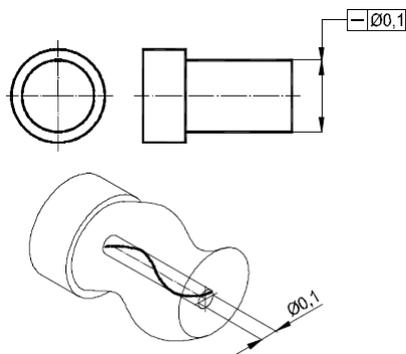


Figura 2.9a: Tolerancia de rectitud.

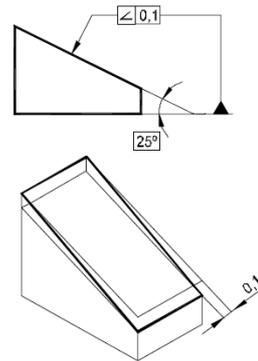


Figura 2.9b: Tolerancia de inclinación.

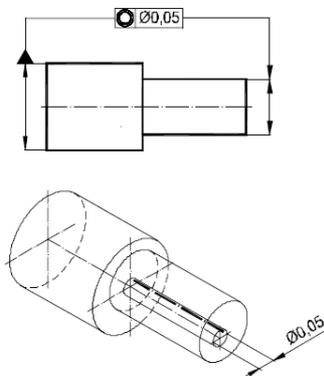


Figura 2.9c: Tolerancia de concentricidad.

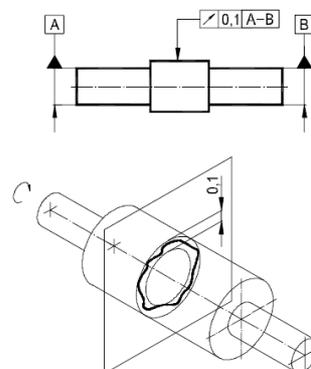


Figura 2.9d: Tolerancia de circularidad radial.

Las tolerancias dimensionales, que hacen referencia a los límites dados para los valores de las distintas dimensiones de las que se compone una pieza o conjunto mecánico se pueden ver en la figura 2.10. Algunas de estas tolerancias están normalizadas dentro de la normativa española UNE 4-040-81, equivalente a la normalización ISO-R-286.

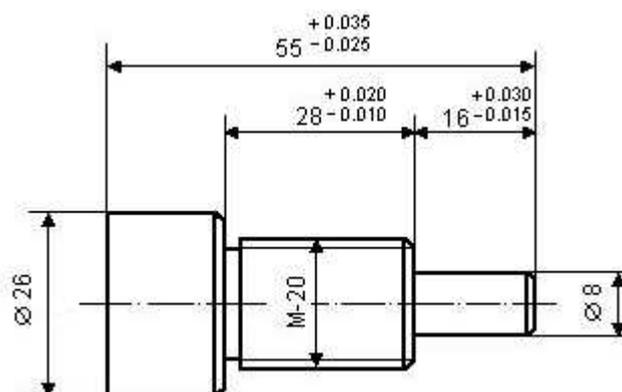


Figura 2.10: Tolerancias dimensionales.

La UNE 1-149 (ISO 8015) "Principio de tolerancias fundamentales" establece el principio de relación entre tolerancias dimensionales y geométricas. En este proyecto nos centraremos en las tolerancias dimensionales y más concretamente en la asignación estadística de los límites de tolerancia superior e inferior de las dimensiones que conforman una pieza o conjunto mecánico.

2.2.2. Definición del problema de asignación estadística óptima de tolerancias dimensionales y valores nominales.

Los cambios producidos en la industria de la manufactura, debidos a la tendencia hacia la globalización, provocaron la búsqueda de la mejora continua de los productos, de su calidad y sobre todo, de la reducción de tiempos de producción así como la necesidad de introducir una serie de nuevas tecnologías acordes a las necesidades.

El diseño robusto de tolerancias es una metodología importante para mejorar la manufactura y la vida útil del producto. Las primeras metodologías para la aplicación de estadística para mejorar la calidad de los productos manufacturados fueron propuestas por el ingeniero y estadista Genichi Taguchi, en 1950, dándose a conocer como el *Método Taguchi*.

La contribución más importante del Dr. Taguchi, fue la aplicación de la estadística y la ingeniería para la reducción de costes y mejora de la calidad en el diseño de productos y los procesos de fabricación. En sus métodos emplean la experimentación a pequeña escala con la finalidad de reducir la variación y descubrir diseños robustos y baratos para la fabricación en serie. Las aplicaciones más avanzadas de los Métodos Taguchi, permiten desarrollar tecnología flexible para el diseño y fabricación de familias de productos de alta calidad, reduciendo los tiempos de investigación, desarrollo y entrega del diseño.

Desde la introducción del diseño robusto a la industria de EE. UU. en los años 80, el acercamiento de los métodos de Taguchi a la Ingeniería de Calidad y el diseño robusto ha recibido la atención de diseñadores, fabricantes, estadísticos y profesionales de la calidad.

El problema de asignación de tolerancias dimensionales y valores nominales a un conjunto de variables se define a continuación. Por simplicidad en adelante se hablará de tolerancias, dando por sentado que se trata de tolerancias dimensionales.

Consideremos que una serie de m variables $\mathbf{Y} = Y_1, Y_2, \dots, Y_m$, son unas características de las que depende el correcto funcionamiento de una pieza o conjunto mecánico. Podemos considerar entonces que esta pieza o conjunto funcionara correctamente si estas variables \mathbf{Y} , son conformes, es decir, están dentro de unos límites de especificación marcados, esto es:

$$Y_j \in [LEI_j, LES_j]$$

Siendo, LEI_j y LES_j , los límites de especificación inferior y superior, respectivamente, de cada una de las variables \mathbf{Y} .

Se asume, que bajo condiciones óptimas de fabricación, la proporción de piezas defectuosas (piezas cuyas variables \mathbf{Y} están fuera de las especificaciones), denotada por p , no debe ser mayor que un pequeño valor α .

Por otro lado, las variables \mathbf{Y} dependen de k variables $\mathbf{X} = X_1, X_2, \dots, X_k$, es decir:

$$\mathbf{Y} = f[X_1, X_2, \dots, X_k]$$

Estas variables \mathbf{X} son características de la pieza y pueden considerarse independientes o podrían considerarse variables correladas. Esta correlación podría ser producto del proceso de fabricación de las piezas.

En el caso de las variables \mathbf{Y} , dado que dependen de las mismas variables \mathbf{X} , es muy probable que exista también dependencia entre ellas.

El problema consiste en encontrar las tolerancias óptimas $\mathbf{t} = t_1, t_2, \dots, t_k$ y los valores nominales óptimos $\mathbf{VN} = VN_1, VN_2, \dots, VN_k$, de cada una de las k variables \mathbf{X} . Los valores óptimos serán aquellos que permitan maximizar las tolerancias obteniendo una proporción de defectuosos menor que α .

Dadas las tolerancias y valores nominales óptimos, los límites de tolerancia de las variables \mathbf{X} serán los siguientes:

$$\begin{aligned} LTI_i &= VN_i - t_i \\ LTS_i &= VN_i + t_i, \quad i = 1, 2, \dots, k \end{aligned}$$

En la figura 2.11 podemos ver un ejemplo. Se trata de un ensamblaje mecánico compuesto por dos piezas definidas por las variables $\mathbf{X} = X_1, X_2, \dots, X_8$.

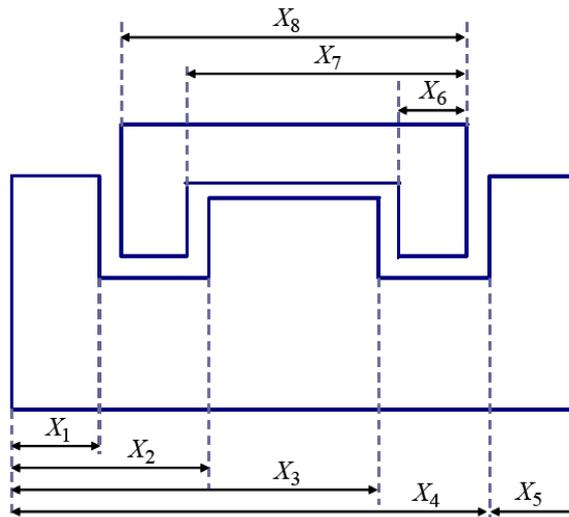


Figura 2.11: Ensamble prismático.

En este caso, es muy probable que las variables X_1, X_2, X_3, X_4, X_5 estén correladas debido al proceso de fabricación. Lo mismo sucede con las variables X_6, X_7, X_8 .

La relación entre las variables \mathbf{Y} con las variables \mathbf{X} se establece mediante la siguiente relación lineal:

$$Y_1 = X_4 + X_5.$$

$$Y_2 = X_2 - X_1 - X_8 + X_7.$$

$$Y_3 = X_7 - X_6 - X_3 + X_2.$$

$$Y_4 = X_4 - X_3 - X_6.$$

Las restricciones de este conjunto corresponden a las siguientes especificaciones de las variables $\mathbf{Y} = Y_1, Y_2, Y_3, Y_4$:

$$Y_1 \leq 127.13.$$

$$Y_2 \geq 0.0076.$$

$$Y_3 \geq 0.0254.$$

$$Y_4 \geq 0.0076.$$

El problema consiste en encontrar las tolerancias y los valores nominales de las variables \mathbf{X} , de manera que se maximicen las tolerancias $\mathbf{t} = t_1, t_2, \dots, t_8$ de \mathbf{X} y se cumplan las especificaciones de las variables \mathbf{Y} con una proporción de defectuosos menor o igual que α .

Resumiendo, las variables a optimizar son los valores nominales y las tolerancias de cada variable X . La función objetivo es maximizar el conjunto de tolerancias $t = t_1, t_2, \dots, t_8$ de X . La restricción es obtener una proporción de defectuosos menor o igual que α . Y, se añadirán restricciones de tolerancias mínimas y límites superior e inferior para los valores nominales, de modo que se pueda resolver adecuadamente el problema.

En el siguiente apartado se describirán brevemente algunos procedimientos ya existentes para la asignación óptima de tolerancias. Estos procedimientos servirán de base para la metodología que se propondrá en este proyecto.

2.3. ASIGNACIÓN ESTADÍSTICA DE TOLERANCIAS CONSIDERANDO VARIABLES INDEPENDIENTES. MÉTODOS ACTUALES.

Las metodologías propuestas por, Chen (1996), Wu et al. (1998), Feng and Balusu (1999), Lee and Tang (2000), Ye and Salustri (2003) y Huang et al. (2005) para el diseño de tolerancias asumiendo la independencia de variables, comparten aspectos en común y nos servirán como punto de referencia para posteriormente entender mejor la asignación de tolerancias asumiendo dependencia de variables.

2.3.1. Metodología utilizada bajo independencia.

En las metodologías propuestas para este tipo de diseño se asume la normalidad de las variables $X = X_1, X_2, \dots, X_k$, por lo que se puede decir que cada variable sigue una distribución:

$$X_i \approx N(\mu_i, \sigma_i^2)$$

Siendo μ_i el valor medio σ_i^2 y la varianza.

Asimismo, los procedimientos que optimizan las tolerancias, asumen que el proceso de fabricación está centrado en el valor nominal, es decir, los valores nominales son iguales a los valores medios, μ_i , de cada variable X_i y se consideran constantes. En cuanto a las tolerancias de cada variable X_i , éstas se expresan como una función de las desviaciones estándar σ_i , mediante el uso del índice de capacidad del proceso. Este método se conoce como la aproximación seis-sigma, propuesto por (Harry and Stewart, 1988; y Hong and Chang, 2002).

Bajo esta suposición y asumiendo tolerancias simétricas con respecto al valor nominal, los límites de tolerancias de las variables X_i pueden ser expresados en términos de un número de desviaciones estándar σ_i de la media.

Asumiendo que el índice de capacidad de la variable X_i es igual a la unidad, $C_p = 1$, entonces las tolerancias se pueden calcular como $t_i = 3\sigma_i$, siendo por tanto los límites de tolerancias los siguientes:

$$X_i \in LTI_i, LTS_i = \mu_i - t_i ; \mu_i + t_i = \mu_i - 3\sigma_i ; \mu_i + 3\sigma_i \quad (2.17)$$

Luego, bajo la hipótesis de independencia de \mathbf{X} , la asignación óptima de las tolerancias consiste en maximizar los valores de t_i , en la ecuación (2.17), sujeta a la restricción de producir una proporción de defectuosos determinada, es decir sujeto a que $p \leq \alpha$, donde α es un valor pequeño, habitualmente $\alpha = 0.0027$.

Esta proporción de defectuosos es calculada, para el caso en el que sólo tengamos una variable Y , como:

$$p \equiv P Y \in LEI, LES \leq \alpha \quad (2.18)$$

Dado que $t_i = 3\sigma_i$, el problema de asignación óptima de tolerancias es expresado como la optimización de las k desviaciones estándar σ_i .

Nótese que las desviaciones estándar óptimas no serán necesariamente los valores de variabilidad del proceso de fabricación, por lo tanto, si los valores óptimos de las desviaciones estándar son mayores que la variabilidad del proceso de fabricación, entonces se podrá tener un índice de capacidad mayor que uno $C_p \geq 1$. De manera opuesta, si el valor óptimo de las desviaciones estándar es menor que la variabilidad del proceso de fabricación, se podrá tener un índice de capacidad menor que uno $C_p \leq 1$, por lo que el proceso sería no apto y debería ser mejorado.

Para verificar la proporción de defectuosos de la ecuación (2.18), se estimará la distribución de la variable mediante la relación $Y = f X_1, X_2, \dots, X_k$. Si esta relación es lineal, de la forma, $Y = a_0 + a_1 X_1 + \dots + a_k X_k$ ó $Y = a_0 + \mathbf{a}'\mathbf{X}$ donde $\mathbf{a} = a_1, a_2, \dots, a_k$, la distribución de la variable Y será $Y \approx N \eta_Y, \sigma_Y^2$, donde:

$$\eta_Y = a_0 + a_1 \mu_1 + \dots + a_k \mu_k \quad (2.19)$$

y, por la independencia de X_i :

$$\sigma_Y = \sqrt{a_1^2 \sigma_1^2 + a_2^2 \sigma_2^2 + \dots + a_k^2 \sigma_k^2} \quad (2.20)$$

Si la relación entre \mathbf{X} e Y no es lineal, se puede aplicar una aproximación mediante la expansión de Taylor. Para este caso, los coeficientes a_1, a_2, \dots, a_k , pasarían a ser igual a la primera derivada de la variable Y respecto a cada variable X_i , esto es:

$$a_i = \frac{\delta Y}{\delta X_i}$$

Estas derivadas son evaluadas para $X_i = \mu_i$.

Cuando la relación sea altamente no lineal, se podrá aplicar series de expansión de Taylor (Evans 1975).

Una vez calculada la distribución de la variable Y , podrá ser calculada la proporción de defectuosos p , de acuerdo a la ecuación (2.18).

Este problema se puede extender a un caso multivariante, donde \mathbf{Y} es un vector de variables, tal que, $\mathbf{Y} = Y_1, Y_2, \dots, Y_j$, con sus correspondientes límites de especificación.

$$Y_j \in [LEI_j, LES_j]$$

Como se puede ver en Cheng (1996) y Lee and Wood (1990).

En este caso, en la mayoría de los métodos mencionados antes, cada variable Y_j es tratada como independiente, por lo que la proporción de no conformidad p será aplicada individualmente a cada una de ellas. En el caso de la metodología propuesta por Lee and Wood (1990) se aplica una corrección para ajustar la proporción de no conformidad deseada α usando una proporción individual α_j para cada variable Y_j , esto es:

$$\alpha_j = 1 - (1 - \alpha)^{1/J}$$

Como se mencionó en el apartado anterior, la suposición de independencia de las variables \mathbf{Y} puede no ser válida puesto que estas dependen de las mismas variables \mathbf{X} , esto es:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \quad (2.21)$$

Resumiendo, los pasos para la asignación óptima de tolerancias bajo la hipótesis de independencia son los siguientes. Para simplificar, se asumirá que $\mu_i = 0$; $i = 1, 2, \dots, k$, también que sólo tenemos una variable Y , y por tanto $\eta_Y = 0$.

1. Inicialización, iteración $r = 0$. Definición del valor inicial de σ_i^r . $i = 1, 2, \dots, k$.
2. Calculo de σ_Y^r mediante el remplazo de los valores σ_i^r en la ecuación (2.20).
3. Calculo de la proporción de piezas no conformes p , según la ecuación (2.18).

$$p = P \left(\mathbf{Y} \notin [LEI, LES] \mid \mathbf{Y} \approx 0, \sigma_Y^{2r} \right)$$

4. Si $\alpha - p > \varepsilon$, donde ε es un pequeño valor, se pasa a la siguiente iteración, $r = r + 1$, asignando un nuevo valor a cada σ_i^r y volviendo al paso 2. De lo contrario, ir al paso 5.
5. Calcular $t_i = 3\sigma_i^r$. Entonces los límites de tolerancias son:

$$LTI_i = \mu_i - t_i \text{ y } LTS = \mu_i + t_i$$

Para resolver este problema de optimización se pueden utilizar diferentes métodos, como por ejemplo, los basados en el gradiente, como los multiplicadores de Lagrange, o métodos evolutivos, como las redes neuronales o los algoritmos genéticos, estos últimos serán presentados y utilizados más adelante en el proyecto.

2.4. ASIGNACIÓN ESTADÍSTICA DE VALORES NOMINALES CONSIDERANDO VARIABLES INDEPENDIENTES. METODOS ACTUALES.

La asignación estadística de valores nominales es también un problema en el diseño de piezas, por ejemplo, en el diseño de circuitos integrados, elementos mecánicos, etc. Autores como por ejemplo Pierluzzi y Rocco, (ver anexo [A.3]), Li y Wu, entre otros, han propuesto algunas metodologías para resolver este problema. Algunas de las metodologías consideran la asignación de valores nominales junto con la asignación de tolerancias, y otras consideran sólo el problema de asignación de valores nominales, manteniendo constante las tolerancias. Es decir los valores $\mathbf{t} = t_1, t_2, \dots, t_k$ son conocidos y constantes. Lo que se quiere asignar son los valores nominales y en consecuencia los límites de tolerancia correspondientes:

$$LTS = VN_i + t_i$$

$$LTI = VN_i - t_i$$

A continuación se describe brevemente la metodología propuesta por Li y Wu, que sigue los mismos principios que el procedimiento para la asignación de tolerancias descrito en el apartado anterior.

2.4.1. Metodología utilizada bajo independencia.

Al igual que en el caso anterior, la variable Y es una característica que asegura el correcto funcionamiento de una pieza o conjunto mecánico.

Esta pieza o conjunto, se podrá considerar correcta o conforme si la variable Y está dentro de unos límites de especificación definidos.

$$Y \in LEI, LES$$

Siendo LEI y LES los límites de especificación inferior y superior respectivamente.

Se asume que bajo condiciones óptimas, la proporción de piezas no conformes p no es mayor que un pequeño valor α que es:

$$p \equiv P Y \in LEI, LES \leq \alpha$$

La variable Y dependerá de una serie de k variables \mathbf{X} , tales que, $\mathbf{X} = X_1, X_2, \dots, X_k$ que definen la pieza o conjunto mecánico, es decir:

$$Y = f [X_1, X_2, \dots, X_k]$$

El problema consiste en encontrar los valores nominales óptimos de cada variable independiente de \mathbf{X} , representados por:

$$\mathbf{VN} = VN_1, VN_2, \dots, VN_k$$

Estos valores nominales son asignados a cada variable X_i teniendo en cuenta el cumplimiento de la proporción de piezas defectuosas.

Al igual que en el caso anterior, se asume normalidad de las variables \mathbf{X} , con la siguiente distribución:

$$X_i \approx N(VN_i, \sigma_i^2)$$

siendo $\sigma_i = t_i/3$, es decir se hace uso también del índice de capacidad.

La resolución del problema es por tanto, similar a la asignación de tolerancias descrito en el apartado anterior. En cada iteración se asignan unos valores nominales, se encuentra la distribución de Y según 2.19 y 2.20 y se calcula la proporción de defectuosos según (2.18). En este caso se define una función objetivo que representa un coste y se busca minimizar dicho coste.

La solución, por tanto, son los valores nominales que permiten obtener el mínimo coste, cumpliendo una proporción de defectuosos igual o menor que α .

2.5. ASIGNACIÓN ESTADÍSTICA DE TOLERANCIAS CONSIDERANDO VARIABLES DEPENDIENTES. METODOS ACTUALES.

González y Sánchez (2008), (ver anexo [A.1]), proponen una metodología para asignar tolerancias a un conjunto de variables dependientes \mathbf{X} . En esta metodología, se asume también la posible dependencia de las variables \mathbf{Y} . Sin embargo, no se considera la asignación de valores nominales, éstos se asumen como valores constantes, especificados a priori.

La metodología que se presentará en este proyecto es una extensión de la metodología propuesta por González y Sánchez (2008). Por este motivo, se describirá con detalle en el capítulo IV, dedicado a la descripción de la metodología que se propone aquí.

3. CAPÍTULO TERCERO: ALGORITMOS GENÉTICOS Y SU USO EN MATLAB

Este capítulo se divide en dos apartados. En el primero se describirán de forma breve los fundamentos teóricos de los algoritmos genéticos y su utilización. En el segundo apartado, se darán las pautas básicas sobre su aplicación como herramienta de optimización usando la toolbox de Matlab "gatool" y se ilustrará con un ejemplo.

3.1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS (AG).

Los Algoritmos Genéticos, desde ahora AG, son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los AG son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagaran en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "superindividuos", cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

El poder de los AG proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el AG encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al AG, tanto en rapidez como en eficacia. El gran campo de aplicación de los AG se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas combinándolas con los AG.

Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a fines de los 60s desarrolló una técnica que permitió incorporarla a un programa. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" tras la publicación de su libro, "*Adaptación en sistemas naturales y artificiales*" publicado en 1975.

Una definición bastante completa de un algoritmo genético puede ser:

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

Introducción a los Algoritmos Genéticos: Carlos A. Coello.

3.1.1. Breve reseña histórica.

Como hemos visto antes, los algoritmos genéticos (AG), fueron inventados en 1975 por John Holland, de la Universidad de Michigan. Los AG son, simplificando, algoritmos de optimización, es decir, tratan de encontrar la mejor solución a un problema dado entre un conjunto de soluciones posibles. Los mecanismos de los que se valen los AG para llevar a cabo esta búsqueda pueden verse como una metáfora de los procesos de evolución biológica propuestos por Charles Darwin en su obra, “El origen de las especies”.

John Holland desde pequeño, se preguntaba cómo logra la naturaleza crear seres cada vez más perfectos. No sabía la respuesta, pero tenía una cierta idea de cómo hallarla, tratando de hacer pequeños modelos de la naturaleza que tuvieran alguna de sus características y ver cómo funcionaban, para luego extrapolar sus conclusiones a la totalidad.

Fue a principios de los 60, en la Universidad de Michigan en Ann Arbor, donde, dentro del grupo Logic of Computers, sus ideas comenzaron a desarrollarse y a dar frutos. Y fue, además, leyendo un libro escrito por un biólogo evolucionista, R. A. Fisher, titulado “La teoría genética de la selección natural”, como comenzó a descubrir los medios de llevar a cabo sus propósitos de comprensión de la naturaleza. De ese libro aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje, y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado. En esta universidad, Holland impartía un curso titulado teoría de sistemas adaptativos. Dentro de este curso, y con una participación activa por parte de sus estudiantes, fue donde se crearon las ideas que más tarde se convertirían en los AG.

Por tanto, cuando Holland se enfrentó a los AG, los objetivos de su investigación fueron dos: imitar los procesos adaptativos de los sistemas naturales y diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

“En lugar de envidiar la naturaleza debemos emularla”

Unos 15 años más adelante, David Goldberg, conoció a Holland, y se convirtió en su estudiante. Goldberg era un ingeniero industrial trabajando en diseño de pipelines, y fue uno de los primeros que trató de aplicar los AG a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complicado como para aplicarle AG, Goldberg consiguió lo que quería, escribiendo un AG en un ordenador personal Apple II. Estas y otras aplicaciones creadas por estudiantes de Holland convirtieron a los AG en un campo con bases suficientemente aceptables como para celebrar la primera conferencia en 1985, ICGA’85.

3.1.2. Descripción del algoritmo de optimización.

Un algoritmo genético consiste en una función matemática o una rutina de software que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación.

Versiones más complejas de algoritmos genéticos generan un ciclo iterativo que directamente toma a la especie (el total de los ejemplares) y crea una nueva generación que reemplaza a la antigua una cantidad de veces determinada por su propio diseño. Una de sus características principales es la de ir perfeccionando su propia heurística en el proceso de ejecución, por lo que no requiere largos períodos de entrenamiento especializado por parte del ser humano, principal defecto de otros métodos para solucionar problemas, como los Sistemas Expertos.

Los algoritmos genéticos son métodos estocásticos de búsqueda ciega. En ellos se mantiene una población que representa a un conjunto de posibles soluciones la cual es sometida a ciertas transformaciones con las que se trata de obtener nuevos candidatos y a un proceso de selección sesgado a favor de los mejores candidatos.

Los algoritmos genéticos se pueden definir como métodos de búsqueda:

- **Ciega:** No necesita tener conocimientos específicos previos del problema, la búsqueda de la solución optimizada se basa exclusivamente en los valores que toma la función objetivo.
- **Codificada:** Nos permite trabajar con cadenas de códigos que representan a los elementos sobre los que vamos a trabajar. Siendo la codificación binaria (0 y 1) la más utilizada.
- **Múltiple:** Permiten la búsqueda simultánea entre un conjunto de candidatos.
- **Estocástica:** Esta combinación aleatoria está relacionada con las fases de selección y transformación.

Los AG trabajan con una población definida que nos permite reducir la posibilidad de alcanzar un falso óptimo local durante su ejecución. Además estos algoritmos se centran únicamente en los resultados de su función objetivo ignorando la información innecesaria.

Los AG nos muestran como se procesan externamente cadenas de códigos, sin embargo, basándonos en el “Teorema fundamental de los algoritmos genéticos” o “Teorema de Holland” podemos ver como lo que procesan internamente son similitudes entre cadenas, de manera que al procesar cada una de las cadenas de la población se procesan a la vez todos los patrones de similitud que contienen, que son muchos más. Es por esta propiedad por la que los AG son mucho más eficaces que otros métodos de búsqueda ciega.

3.1.3. Estructura de los algoritmos genéticos.

Basándose en las teorías evolutivas, los algoritmos genéticos siguen un esquema básico de selección y búsqueda. Partiendo de una población inicial, normalmente aleatoria, de n individuos, se seleccionan aquellos cuyas propiedades les otorguen una mayor probabilidad de ser seleccionados como individuo mejor valorado, formándose una nueva población intermedia de n individuos. De esta nueva población se extraerán a $m \leq n$ individuos que formaran el grupo de los progenitores. Sobre estos progenitores se aplicarán transformaciones mediante los operadores genéticos de cruce y mutación para generar m nuevos individuos que constituyen la descendencia. Para formar la nueva población de n individuos se realizará la fase de reemplazo, por la cual se eligen individuos de la población intermedia y de la descendencia.

Este bucle de selección-cruce-mutación-reemplazo se repite un número determinado de generaciones, hasta alcanzar la solución óptima. Ver figura 3.1.

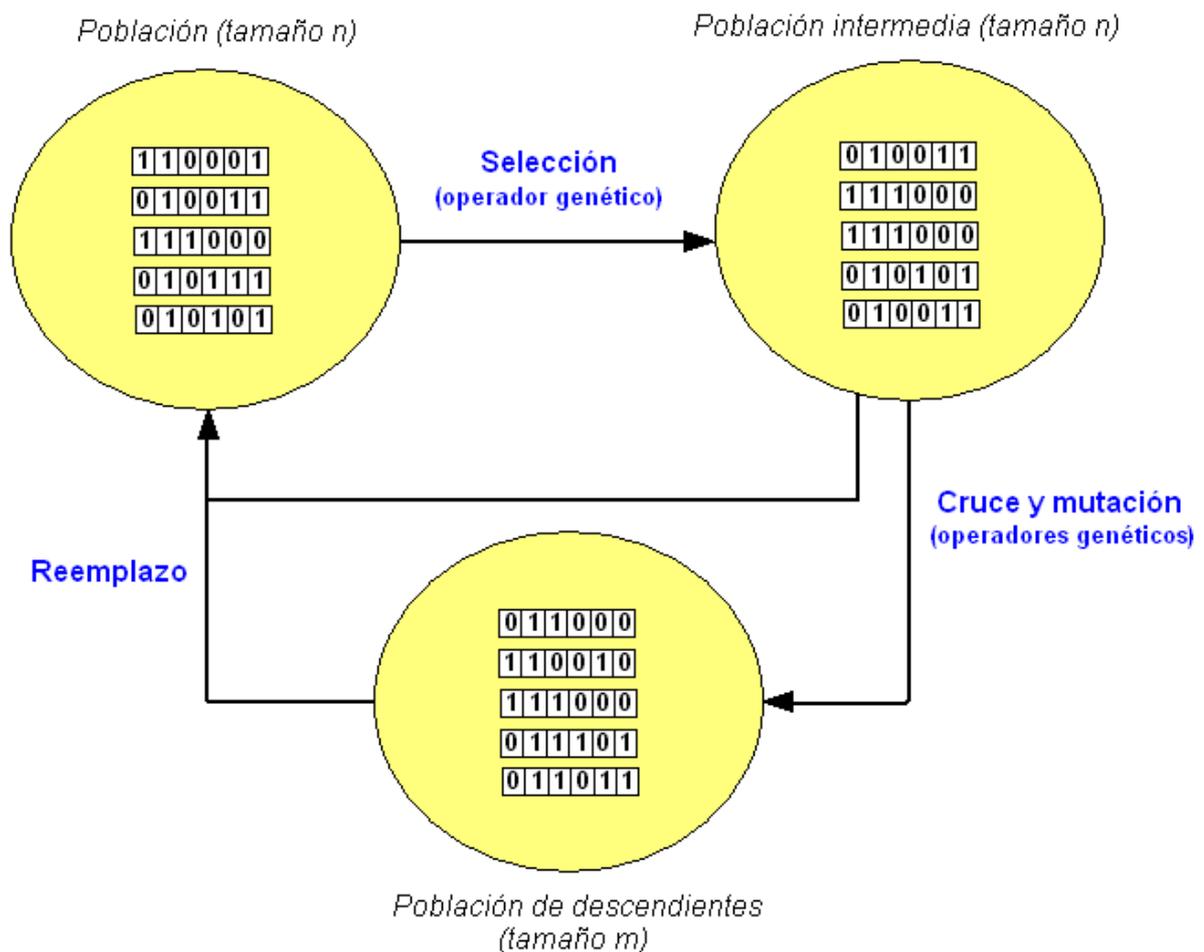


Figura 3.1: Bucle selección-cruce-mutación-reemplazo.

3.1.3.1. Codificación.

Los algoritmos genéticos requieren que cada número entero o decimal, se codifique en una *cromosoma*. Cada cromosoma tiene varios genes, que corresponden a sendos parámetros del problema. Para poder trabajar con estos genes en el ordenador, es necesario codificarlos en una *cadena*, es decir, una ristra de símbolos (números o letras) que generalmente va a estar compuesta de ceros y unos, que son los componentes de la codificación binaria.

Codificación binaria

Hay otras codificaciones posibles, usando números naturales o letras del alfabeto; sin embargo, uno de los resultados fundamentales en la teoría de algoritmos genéticos, *El teorema de los esquemas*, afirma que la codificación óptima es aquella que tiene un alfabeto de cardinalidad dos, siendo este el caso de la codificación binaria.

Es importante una adecuada representación binaria de todas las posibles soluciones, cada solución debe tener una única representación binaria y será necesario especificar el procedimiento con el que se hace corresponder cada punto del dominio del problema con un código, o utilizando terminología de algoritmos evolutivos, se debe especificar el mecanismo de paso del genotipo (código) a los fenotipos (Valor real) y viceversa.

La figura 3.2 muestra un ejemplo de codificación para un problema de dos variables x_1 y x_2 .

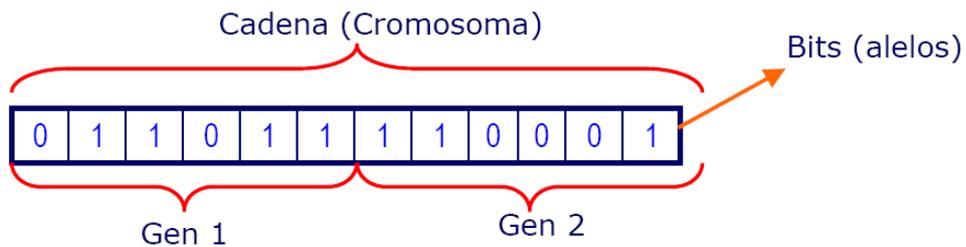


Figura 3.2: Codificación binaria de dos variables.

En este caso las variables han sido representadas mediante 6 bits, por tanto, cada una puede tomar 2^6 valores distintos.

Codificación binaria de números enteros

Cuando, por ejemplo, el dominio de un problema pueda representarse mediante números enteros, cada variable representada por una longitud $L=5$ bits puede tomar un total de $2^5=32$ valores distintos. De esta forma, la representación binaria se realizara como se puede ver en la figura 3.3.

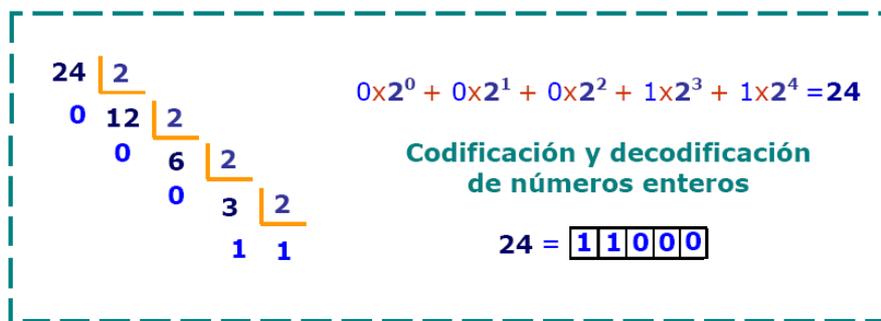


Figura 3.3: Codificación binaria de un número entero.

Codificación binaria de números decimales

Cuando el dominio del problema ha de representarse mediante números decimales, la longitud L , se calcula, transformando el intervalo de soluciones a un intervalo discreto y teniendo en cuenta la precisión que se quiere tener. De esta forma, la distancia entre puntos consecutivos del intervalo, debe ser menor o igual que el valor de la precisión especificada m .

$$X_{\min}, X_{\max} \leq m$$

Para ver claramente como calcular la longitud L de las variables, planteamos un ejemplo en el que para soluciones comprendidas en el intervalo $[0,1]$ con precisión $m = 0.005$, la longitud L de la representación binaria es calculada como:

$$\frac{X_{\max} - X_{\min}}{2^L - 1} < m \Rightarrow L = \log_2 \left(1 + \frac{X_{\max} - X_{\min}}{m} \right) \quad (3.1)$$

En este caso obtenemos un valor de $L = 8$, siendo: $X_{\min} = 000...0$ y $X_{\max} = 111...1$. Por lo que podremos representar $2^8 = 256$ valores de la variable comprendidos en el intervalo X_{\min}, X_{\max} , obteniendo una precisión mayor a la especificada:

$$\frac{X_{\max} - X_{\min}}{2^L - 1} < m \Rightarrow \frac{1-0}{2^8 - 1} < 0.005 \Rightarrow 0.0039 < 0.005$$

La forma de decodificar los números decimales se realizará según se indica en este ejemplo:

Dado el valor decimal $X = 0.7294$, cuya representación binaria es $v = 10111010$, el valor de X se ha obtenido mediante:

$$X = X_{\min} + bin2dec \ v \cdot \left(\frac{X_{\max} - X_{\min}}{2^L - 1} \right) \quad (3.2)$$

Donde *bin2dec* . significa pasar del valor binario v , al valor decimal. En este caso:

$bin2dec(10111010) = 186$, por lo tanto;

$$X = 0 + 186 \cdot \left(\frac{1-0}{2^8 - 1} \right) = 0.7294$$

Una buena codificación debe implicar que pequeños cambios en el valor real, signifiquen pequeños cambios en el código que lo representa. El código binario, puede en algunos casos no cumplir estrictamente esta propiedad, pero es lo suficientemente aceptable para la resolución de este proyecto. Para una mayor precisión en el uso de AG se recomienda el estudio del código gray que se ajusta mejor a esta propiedad.

3.1.3.2. Creación de la población inicial.

La inicialización de la población determina el proceso de creación de los individuos para el primer ciclo del algoritmo. Normalmente, la población inicial se forma a partir de individuos creados aleatoriamente. En el caso de disponer de conocimiento específico del problema se podría elegir, por ejemplo, una población inicial próxima al óptimo.

Conviene que la población inicial contenga la mayor cantidad posible de puntos factibles. En ocasiones es difícil obtener puntos factibles y se deberá comenzar a iterar con algoritmos genéticos una cantidad muy reducida de ellos. En este caso no es conveniente usar un tamaño de población muy pequeño, dado que perjudica gravemente al requisito de diversidad, ni tampoco introducir una gran cantidad de puntos no factibles, ya que se reduce la eficiencia del AG.

El tamaño de la población suele estar comprendido entre 50-100 individuos, aunque un tamaño pequeño de población puede concentrar la búsqueda en zonas pequeñas del espacio de soluciones.

3.1.3.3. Función de evaluación y función de aptitud.

Se deben definir las funciones de evaluación y aptitud más apropiadas para el problema, teniendo en cuenta que para evaluar estas funciones, será necesario decodificar los puntos.

Generalmente, se suele definir como función de evaluación la misma función objetivo, pero puede ser que la función objetivo sea muy compleja, o que no proporcione un valor numérico y, por lo tanto, en esos casos es necesario definir una función de evaluación diferente.

La función de aptitud (fitness), no es más que la función objetivo de nuestro problema de optimización. Una característica que debe tener esta función de aptitud es la de castigar las malas soluciones y premiar las buenas, para valorar qué tan buenos son los individuos.

Se deben tomar siempre valores positivos ya que los AG maximizan siempre la función de aptitud, por tanto, se admite por defecto que se maximiza el objetivo. En ocasiones resulta más apropiado plantear el problema como uno de minimización, en ese caso es inmediato transformar estos problemas en otros de maximización sin más que cambiarle el signo a la función.

Una forma de construir la función de aptitud es desplazando y escalando convenientemente la función de evaluación. El desplazamiento tiene como finalidad principal hacer que la función de aptitud devuelva valores positivos. El procedimiento más usado para hacer positivas las evaluaciones es el de la ventana de desplazamiento que consiste en desplazar las evaluaciones como sigue:

$$Fitness = F - Evaluación$$

donde F es una cota superior para las evaluaciones, esto es, $Fitness \geq F - Evaluación$.

No es habitual conocer a priori el rango de valores que va a tomar la función de evaluación, por lo que es preferible ir adaptando dinámicamente el valor de F del siguiente modo:

$$Fitness = F_{max} - Evaluación$$

donde F_{\max} es la máxima aptitud de cualquier individuo evaluado hasta la iteración inmediatamente anterior, $t - 1$.

Es conveniente ir actualizando el valor de F_{\max} cada cierto número de generaciones con el fin de evitar una posible pérdida de diversidad. Una forma de controlar la diversidad de las aptitudes es obteniendo la función de aptitud mediante el escalonado de la función de evaluación.

3.1.3.4. Selección.

El proceso de selección en los algoritmos genéticos escoge individuos para la reproducción. La selección está basada en la aptitud de los individuos: individuos más aptos tienen mayor probabilidad de ser escogidos para la reproducción.

Este primer operador genético consiste en seleccionar de la población inicial los individuos con mejores aptitudes, por lo que no supondrá la aparición de nuevos individuos, sino que formará una población intermedia de individuos que poseen las mejores aptitudes para que produzcan mayor número de descendientes tras aplicar, posteriormente, los operadores de cruce y mutación.

Los métodos de selección pueden ser estocásticos o deterministas, los primeros son los más usados en la práctica y los segundos están en desuso, entre otros motivos porque van en contra de la filosofía de los AG.

Procedimientos de selección.

Métodos directos: Estos métodos no son estocásticos y no dan la oportunidad de que los peores individuos sean seleccionados. Se selecciona un subconjunto de individuos de la población siguiendo un criterio fijo.

Métodos aleatorios simples o equiprobable: Mediante estos métodos se asignan a todos los elementos de la población base las mismas probabilidades de ser seleccionados.

Métodos estocásticos: Estos métodos asignan probabilidades de selección o puntuaciones a los elementos de la población base en función de su aptitud. Existen muchos mecanismos de muestreo estocásticos los más utilizados son los siguientes.

- Método por sorteo.
- Método universal o por ruleta.
- Método jerárquico.
- Método por torneos.

3.1.3.5. Cruce.

Es el operador genético fundamental de los algoritmos genéticos. Realiza la función de explotación de las soluciones, es decir, intenta potenciar los mejores individuos de la población.

Los operadores de cruce son el arquetipo de operadores de recombinación: actúan sobre parejas de individuos y normalmente originan otro par de individuos que combinan características de los progenitores. El cruce no modifica los valores de los bits de la pareja de individuos que se cruzan, no es posible por tanto, obtener nuevos bits.

Dado que en los AG los individuos están representados a través de cadenas binarias, el cruce se lleva a cabo por intercambio de segmentos. Para realizar el cruce de individuos se define una probabilidad de cruce p_c . Luego, se asigna aleatoriamente a cada individuo un valor " b " comprendido en el intervalo $0,1$ de manera que si $b < p_c$ se realiza un cruce con ese individuo.

Por ejemplo: si tenemos una probabilidad de cruce $p_c = 0.6$ y el tamaño de la población es 100, el valor medio de individuos que se cruzarán es 60. Normalmente se asignan valores de p_c entre 0.6 y 0.95. La elección del individuo que se cruzará con otro también es aleatoria.

Procedimientos de cruce.

Cruce monopunto: El cruce entre una pareja de individuos se realiza cortando las cadenas en un punto. La selección del punto de cruce se elige de forma aleatoria.

Cruce multipunto: Se generan aleatoriamente varios puntos de corte en la cadena y se cruzan dos individuos. Este cruce mejora la capacidad de procesamiento de esquemas, pero acostumbra a perder velocidad de convergencia.

Cruzamiento uniforme: Para cada bit del primer hijo se decide, con cierta probabilidad, de que progenitor heredará el valor para esa posición. El segundo hijo recibe el correspondiente valor para esa posición del otro progenitor. Dado que el cruce uniforme intercambia bits (alelos) en lugar de segmentos de la cadena, se usa para combinar atributos específicos, con independencia de la posición en que han sido codificados. En algunos problemas esta capacidad compensa el riesgo adicional que se introduce al descomponer bloques constructivos. No obstante se recomienda utilizar el cruce uniforme sólo en casos concretos en los que haya motivos fundados para hacerlos.

3.1.3.6. Mutación.

La mutación es un operador genético de importancia teórica secundaria, lo que se manifiesta en su baja probabilidad de aplicación en comparación con el cruce. Esto no significa que en la práctica sea un operador imprescindible, de hecho se ha comprobado que un AG puede funcionar sin realizar cruces, pero no sin realizar mutaciones. El cruce puede originar pérdida de alelos (incesto). La mutación evita este problema.

Los operadores de mutación son operadores de *exploración*, ya que permiten mover el espacio de soluciones a nuevas zonas de búsqueda. La mutación genera alteración en los bits de un individuo, es decir, consiste en modificar alguno de los bits de un individuo o el total de ellos. Por tanto, la mutación sí permite obtener nuevos bits o alelos.

Inicialmente es importante explorar distintas zonas del espacio de soluciones, para ello la probabilidad de mutación debe de ser alta. Esto hace aleatoria la búsqueda. Luego, es importante explotar las mejores soluciones, esto significa probabilidad de cruce alto y probabilidad de mutación baja. Es importante tener en cuenta que probabilidades de mutación muy bajas pueden llevar a óptimos locales.

Para realizar la mutación se define, para cada bit, una probabilidad de mutación p_m . Luego, se asigna aleatoriamente un número "c" entre $[0,1]$ a cada gen y si $c < p_m$, entonces se aplica mutación, es decir el bit cambio de valor. En la figura 3.4 se muestra un ejemplo.



Figura 3.4: Mutación.

3.1.3.7. Reemplazo.

Es el proceso por el cual se eligen los individuos de la población intermedia y de la población de descendientes (después del cruce y mutación), que formarán la nueva población por sustitución de la inicial. Hay que recordar que si se parte de n individuos en la población inicial, se seleccionan para la población intermedia n individuos y de ésta se generan m descendientes. De manera que se tienen en total $n+m$ individuos, de los cuales sólo n formarán parte de la nueva población. Ver figura 3.5.

El reemplazo puede llevarse acabo de diferentes maneras:

Reemplazo al vuelo o inmediato: Los m descendientes sustituyen directamente a sus m progenitores sin ninguna consideración previa.

Reemplazo por factor de llenado: Los m descendientes sustituyen a aquellos individuos de la población intermedia a los que más se asemejan.

Reemplazo por inserción: Este reemplazo se puede realizar de dos maneras:

- En la primera, se tiene un número de descendientes mayor que el tamaño de la nueva población, por lo que, se seleccionan únicamente descendientes para formar la nueva población. Esto provoca que un individuo no pueda subsistir más de una generación.
- En la segunda opción, el número de descendientes es inferior al tamaño de la nueva población, por tanto, todos los descendientes pasan a formar parte de la nueva población junto con los mejores individuos de la población intermedia.

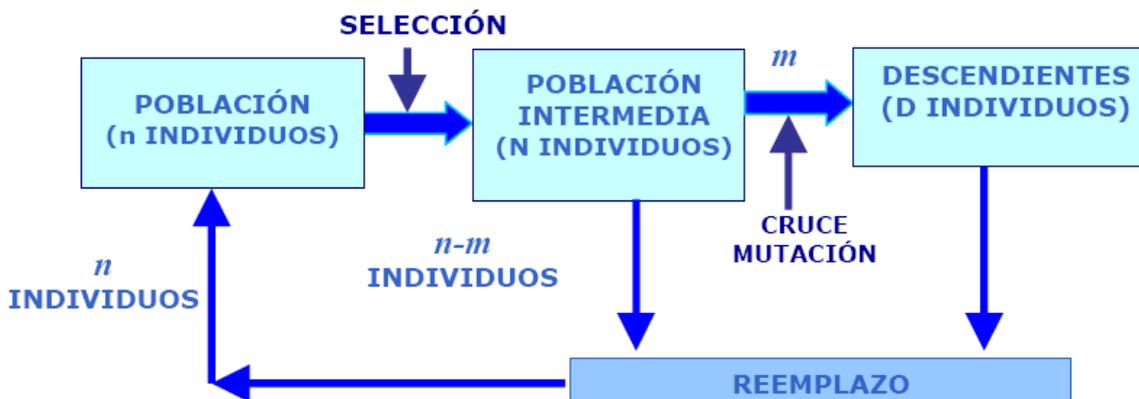


Figura 3.5: Reemplazo.

3.1.3.8. Criterio de parada.

El criterio de parada más habitual es utilizando un número máximo de iteraciones. Este criterio no siempre garantiza buenos resultados, ya que podría no haberse alcanzado el óptimo.

Otro criterio de parada está basado en el grado de homogenización de la población, el cual se puede verificar mediante la cantidad de genes que han convergido. Por ejemplo, se puede parar el algoritmo si converge el 95% del total de genes. Un gen ha convergido cuando cierto porcentaje de la población (por ejemplo el 90%) tiene los mismos alelos en dicho gen.

Otro criterio de parada es también definir un valor T , de manera que si en cierto número de iteraciones no se mejora la solución, un valor menor que T , entonces se termina.

3.1.4. Ejemplo de implementación de un algoritmo genético.

Para una mejor comprensión de lo que es y para lo que sirve un algoritmo genético vamos a realizar un ejercicio muy sencillo que nos servirá para familiarizarnos con esta metodología. Con este ejercicio se pretende mostrar que la aplicación de algoritmos genéticos está al alcance de todos y no es preciso tener grandes conocimientos matemáticos.

Ejercicio: Funcionamiento de un algoritmo.

Vamos a partir de la función:

$$f(x) = x^2$$

Objetivo:

El objetivo de este ejercicio es encontrar el valor de x que haga que la función $f(x)$ alcance su valor máximo, teniendo en cuenta las siguientes restricciones.

- $x \in [1, 31]$.
- $x \in \mathbb{Z}$, siendo \mathbb{Z} el conjunto de los números enteros.

Es evidente que para la resolución de este ejercicio no haría falta la aplicación de algoritmos genéticos, ya que, obviamente el máximo se tiene para $x = 31$, donde $f(31) = 961$. Pero la sencillez del ejercicio permitirá que el algoritmo sea más fácil de entender.

Codificación:

Lo primero que tenemos que hacer es codificar las posibles soluciones, valores de x , que podemos tener. Para ello aplicando la codificación binaria para números enteros que vimos anteriormente, siendo la longitud del código de 5 bits, ya que son los necesarios para definir el máximo valor posible que puede tomar x , por lo que un posible valor de x es:

$$01011$$

Donde, decodificando el código podemos decir que:

$$x = 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 0 + 8 + 0 + 2 + 1 = 11$$

Población inicial:

A cada valor de x en codificación binaria se le denominará individuo, por lo que a una colección de estos individuos se le denominará población y al número de individuos, tamaño de la población.

Una vez codificada las posibles soluciones necesitamos escoger un tamaño de población, en este caso de 6 individuos, y crear una población inicial. Ver tabla 3.1. Esta población inicial va ser creada al azar entre las posibles soluciones, obteniendo:

Tabla 3.1: Población inicial.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	VALOR x	VALOR f(x)
1	0 1 1 0 0	12	144
2	1 0 0 1 0	18	324
3	0 1 1 1 1	15	225
4	1 1 0 0 0	24	576
5	1 1 0 1 0	26	676
6	0 0 0 0 1	1	1

Observamos que el mejor individuo en la población inicial es el número 5 con $f(x) = 676$.

Selección:

Una vez definida la población inicial, este primer operador genético consiste en seleccionar de dicha población los individuos con mejores aptitudes. Para ello, como se ve en la tabla 3.2, se hará competir a los individuos entres sí de manera aleatoria mediante un torneo entre dos, el mejor generará dos copias y el peor se desechará.

Tabla 3.2: Selección.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	VALOR x	VALOR f(x)	PAREJA DE TORNEO
1	0 1 1 0 0	12	144	6
2	1 0 0 1 0	18	324	3
3	0 1 1 1 1	15	225	2
4	1 1 0 0 0	24	576	5
5	1 1 0 1 0	26	676	4
6	0 0 0 0 1	1	1	1

Tras realizar el proceso de selección, en la tabla 3.3, podemos ver la población intermedia.

Tabla 3.3: Población intermedia.

Nº INDIVIDUO	CODIFICACIÓN BINARIA
1	0 1 1 0 0
2	0 1 1 0 0
3	1 0 0 1 0
4	1 0 0 1 0
5	1 1 0 1 0
6	1 1 0 1 0

Cruce:

Tras realizar la selección y definir la población intermedia, este operador genético realiza la función de *explotación* de las soluciones, es decir, intenta potenciar los mejores individuos de la población intermedia. Para ello como se puede ver en la tabla 3.4, se realizará un cruce monopunto, que consistirá en, dados dos individuos pareja se establece un punto de corte aleatorio comprendido entre 1 y 4, (que es el intervalo que queda al quitarle la unidad a la longitud del individuo), y se repartirán los bits formando dos nuevos individuos denominados hijos.

Tabla 3.4: Cruce.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	PAREJA DE CRUCE	PUNTO DE CRUCE
1	0 1 1 0 0	5	1
2	0 1 1 0 0	3	3
3	1 0 0 1 0	2	3
4	1 0 0 1 0	6	1
5	1 1 0 1 0	1	1
6	1 1 0 1 0	4	1

Para una mayor comprensión realizaremos, a modo de ejemplo, el cruce monopunto de los individuos 2 y 3, en este caso el punto de corte obtenido de forma aleatoria ha sido el 3 por lo que tenemos que:

Punto de corte=3.

<i>Padre 1</i>		<i>Padre 2</i>
<i>Individuo 2</i>	↔	<i>Individuo 3</i>
011 00		100 10
 <i>Hijo 1</i>		 <i>Hijo 2</i>
01110		10000

Por lo que la población que queda tras realizar el cruce se puede ver en la tabla 3.5.

Tabla 3.5: Población tras el cruce.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	VALOR DE x	VALOR DE F(x)
1	0 1 0 1 0	10	100
2	1 1 1 0 0	28	784
3	0 1 1 1 0	14	196
4	1 0 0 0 0	16	256
5	1 1 0 1 0	26	676
6	1 0 0 1 0	18	324

Tras aplicar los operadores genéticos de selección y cruce podemos ver como el mejor individuo en la población tras el cruce es el número 2 con $f(x) = 784$.

Comparando este valor con el obtenido en la población inicial $f(x) = 676$, podemos ver como los individuos después de la selección y el cruce son mejores. Por otro lado, si realizamos la media $f(x)_{med} = 324.3$ de la población inicial y la comparamos con la media $f(x)_{med} = 389.3$ de la población tras el cruce, vemos como el conjunto de los individuos tras la selección y el cruce es mejor que antes de la aplicación de estos operadores genéticos.

Mutación:

Para aplicar este operador genético, definimos para cada bit de nuestra población, una probabilidad de mutación $p_m = 5\%$. Luego, se asigna aleatoriamente un número "c" entre $[0,1]$ a cada bit, esto significa, que en promedio, siendo $5 \times 6 = 30$, el número total de bits que mutaran es de $30 \times 0.05 = 1.5 \text{ bits}$. Por lo tanto, los bits que verifiquen $c < p_m$ mutarán, es decir el bit cambiara de valor.

Para simplificar los resultados, como se puede ver en la tabla 3.6, diremos que los bits de cambio se producen en el cuarto bit del individuo 1, en el tercer bit del individuo 5 y el tercer y quinto bit del individuo 6.

Tabla 3.6: Proceso de mutación.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	MUTACIÓN	CODIFICACIÓN NUEVA
1	0 1 0 1 0	⇒	0 1 0 0 0
5	1 1 0 1 0		1 1 1 1 0
6	1 0 0 1 0		1 0 1 1 1

En la tabla 3.7 vemos los resultados tras la mutación.

Tabla 3.7: Población tras la mutación.

Nº INDIVIDUO	CODIFICACIÓN BINARIA	VALOR DE x	VALOR DE F(x)
1	0 1 0 0 0	8	64
2	1 1 1 0 0	28	784
3	0 1 1 1 0	14	196
4	1 0 0 0 0	16	256
5	1 1 1 1 0	30	900
6	1 0 1 1 1	23	529

Comparando los valores obtenidos con el valor tras el cruce vemos como ha mejorado considerablemente siendo tras la mutación, el mejor individuo el número 5 con $f(x) = 900$, y una media de $f(x)_{med} = 454.8$.

Los algoritmos genéticos no nos garantizan la obtención del valor óptimo, pero viendo los resultados que tenemos podemos afirmar que estamos muy cerca del óptimo por lo que podemos considerar al individuo número 5 como óptimo. Esta decisión se puede tomar ya que en este ejercicio no hemos tenido en cuenta ninguna restricción en el criterio de parada.

Remplazo:

Por último tras realizar tantas veces como sea necesario, o como indique el criterio de parada, el bucle de, selección, cruce y mutación, remplazamos mediante el método de remplazo inmediato, como se puede ver en la figura 3.8, la población inicial por la nueva población que contiene el valor óptimo obtenido,

Tabla 3.8: Nueva población.

Nº INDIVIDUO	CODIFICACIÓN BINARIA INICIAL	VALOR x	VALOR f(x)	CODIFICACIÓN BINARIA FINAL	VALOR x	VALOR F(x)
1	0 1 1 0 0	12	144	0 1 0 0 0	8	64
2	1 0 0 1 0	18	324	1 1 1 0 0	28	784
3	0 1 1 1 1	15	225	0 1 1 1 0	14	196
4	1 1 0 0 0	24	576	1 0 0 0 0	16	256
5	1 1 0 1 0	26	676	1 1 1 1 0	30	900
6	0 0 0 0 1	1	1	1 0 1 1 1	23	529

Criterio de parada:

En este ejercicio no se ha considerado ningún criterio de parada específico, pero, si el resultado obtenido tras la mutación no hubiera sido considerado como válido, repetiríamos el bucle de selección, cruce y mutación durante tantas iteraciones como indicara el criterio de parada, o hasta que los resultados alcancen una homogeneidad definida con anterioridad.

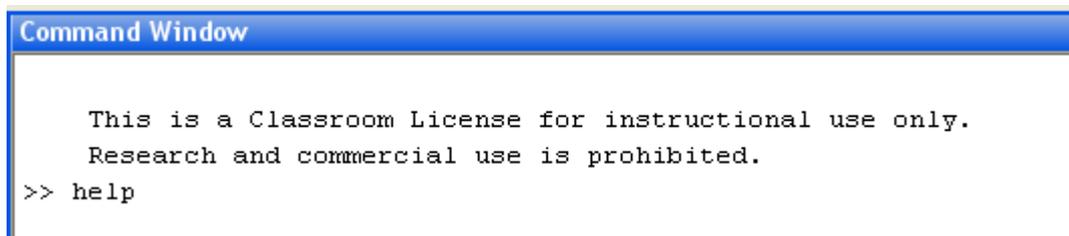
3.2. ALGORITMOS GENÉTICOS CON MATLAB.

En este apartado, se pretende dar una visión general de las posibilidades que ofrece el entorno de trabajo MATLAB para la resolución de algoritmos genéticos. Se recomienda consultar un manual especializado o la autoayuda “help” del entorno, tantas veces como sea preciso para su mayor comprensión.

3.2.1. Toolbox de algoritmos genéticos en MATLAB.

Para el primer contacto con MATLAB, como ya comentamos anteriormente, una posible forma de conocer todas las posibilidades que nos ofrece el programa respecto a la resolución de algoritmos genéticos es, acudir al comando de autoayuda “help” del entorno desde su página de inicio.

Desde la pantalla principal de MATLAB, normalmente dividida en tres ventanas, escribimos “help” en la ventana “Comand Windows”. Ver figura 3.6.



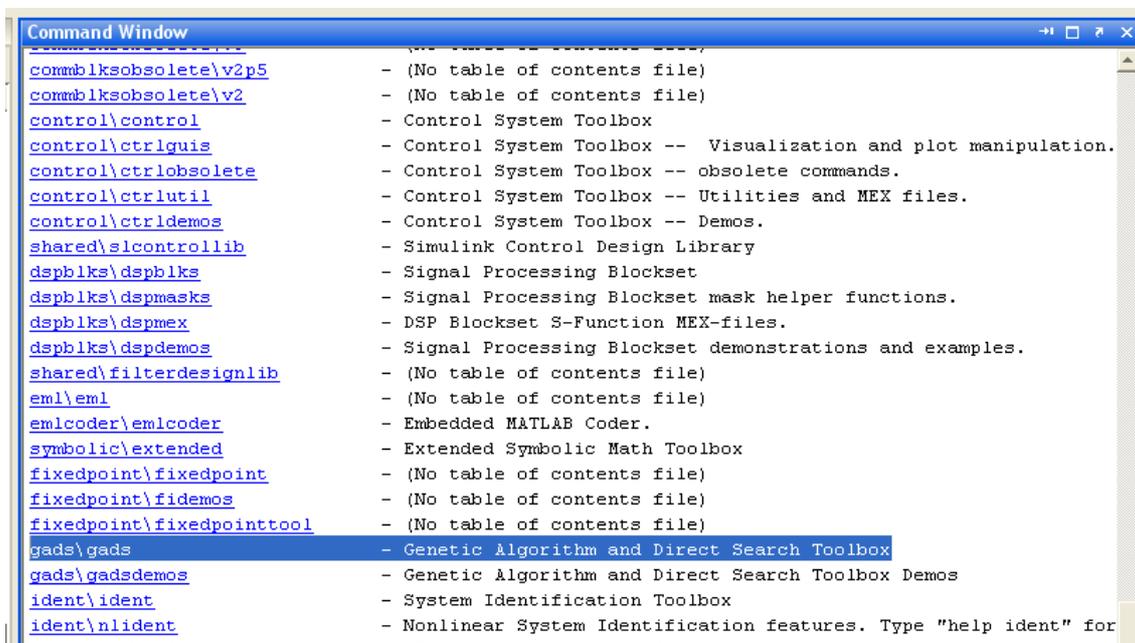
```

Command Window

This is a Classroom License for instructional use only.
Research and commercial use is prohibited.
>> help
  
```

Figura 3.6: Comando help en Comand Windows.

Una vez ejecutado este comando, nos aparecerán todas las posibles aplicaciones de MATLAB con su correspondiente comando de aplicación, como se muestra en la figura 3.7.



```

Command Window

commbksoobsolete\v2p5 - (No table of contents file)
commbksoobsolete\v2 - (No table of contents file)
control\control - Control System Toolbox
control\ctrlquis - Control System Toolbox -- Visualization and plot manipulation.
control\ctrlobsolete - Control System Toolbox -- obsolete commands.
control\ctrlutil - Control System Toolbox -- Utilities and MEX files.
control\ctrldemos - Control System Toolbox -- Demos.
shared\slcontrollib - Simulink Control Design Library
dspblks\dspblks - Signal Processing Blockset
dspblks\dspmasks - Signal Processing Blockset mask helper functions.
dspblks\dspmex - DSP Blockset S-Function MEX-files.
dspblks\dspdemos - Signal Processing Blockset demonstrations and examples.
shared\filterdesignlib - (No table of contents file)
eml\eml - (No table of contents file)
emlcoder\emlcoder - Embedded MATLAB Coder.
symbolic\extended - Extended Symbolic Math Toolbox
fixedpoint\fixedpoint - (No table of contents file)
fixedpoint\fidemos - (No table of contents file)
fixedpoint\fixedpointtool - (No table of contents file)
gads\gads - Genetic Algorithm and Direct Search Toolbox
gads\gadsdemos - Genetic Algorithm and Direct Search Toolbox Demos
ident\ident - System Identification Toolbox
ident\nident - Nonlinear System Identification features. Type "help ident" for
  
```

Figura 3.7: Aplicaciones de MATLAB.

En la figura 3.7 podemos ver que el comando “gads/gads” hace referencia a “Genetic Algorithm and Direct Search Toolbox”, dentro de este, nos centraremos en el comando “gatool”, que se encarga de la optimización de funciones mediante algoritmos genéticos.

Al igual que antes, para acceder directamente a la toolbox de algoritmos genéticos desde la página principal de MATLAB, escribimos el comando “gatool” en la ventana “Comand Windows” como podemos ver en la figura 3.8. Otra forma más directa de llegar a esta herramienta de optimización desde la pantalla inicial de MATLAB, es como se puede ver en la figura 3.9, mediante la secuencia:

Start > Toolboxes > Genetic Algorithm and Direct Search Toolbox > Genetic Algorithm Tool

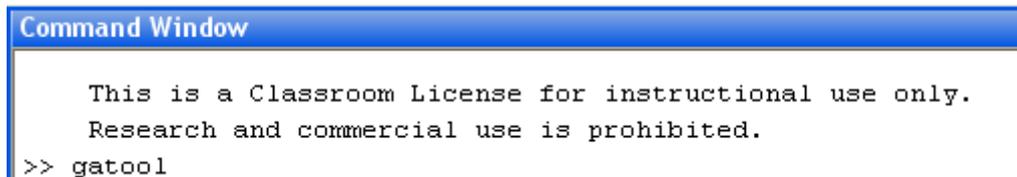


Figura 3.8: Comando “gatool” en Comand Window.

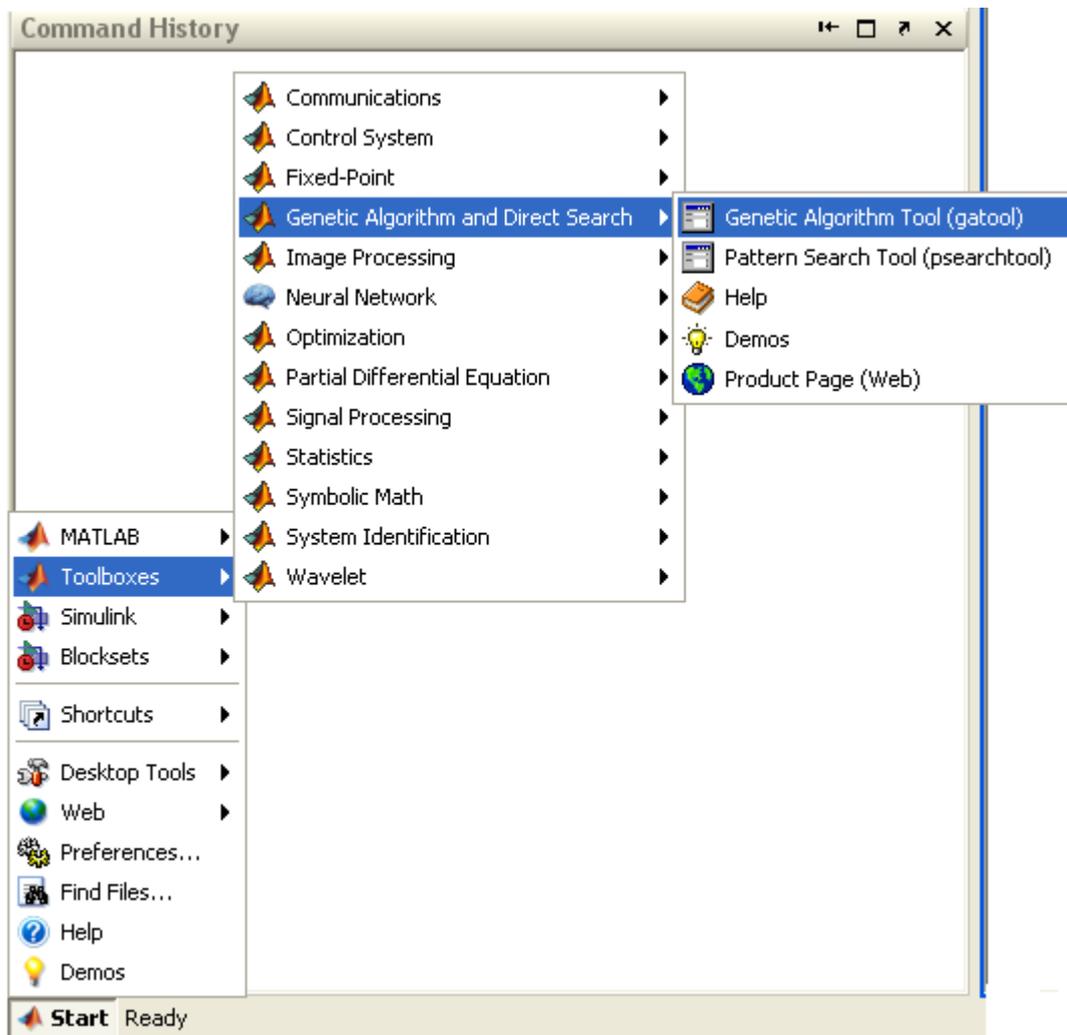


Figura 3.9: Secuencia directa para acceder al comando “gatool”.

Ejecutamos el comando y nos aparece la ventana “Optimization Toll”, como se puede ver en la figura 3.10, donde introduciremos los parámetros necesarios para la resolución del algoritmo.

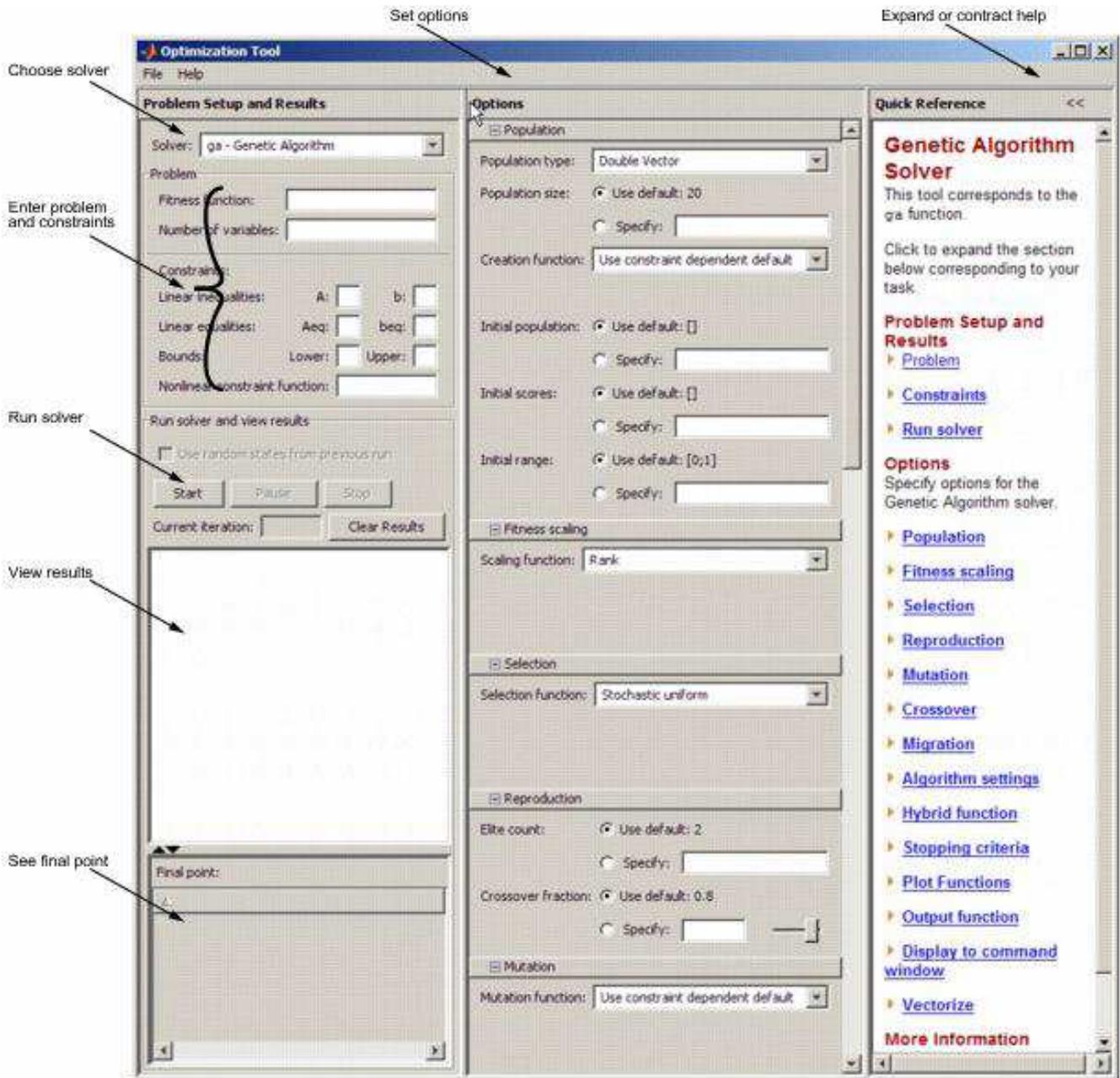


Figura 3.10: Pantalla “Optimización Tool”.

En esta pantalla se definirán las características del problema que queremos optimizar mediante la aplicación de algoritmos genéticos en MATLAB.

3.2.2. Ejemplo de implementación.

Para una mayor comprensión del uso de la herramienta, se va a realizar paso por paso un ejemplo de implementación de algoritmos genéticos en MATLAB.

Supongamos, que la función que queremos optimizar es:

$$f(x_1, x_2) = x_1^2 - 2x_1x_2 + 6x_1 + x_2^2 - 6x_2$$

Los algoritmos genéticos aplicados en MATLAB, como proceso de optimización, minimizan la función hasta conseguir un valor óptimo.

En primer lugar creamos un archivo nuevo, al que llamaremos “Ejemplo” para poder llamar a la función cuando se aplique la toolbox de MATLAB, para ello procedemos como sigue:

1. En la pantalla principal de MATLAB, seleccionamos File-New.
2. Seleccionamos “M-File” y se nos abre un editor, como podemos ver en la figura 3.11.

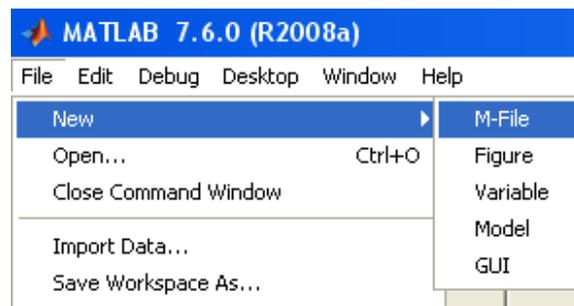


Figura 3.11: Creación de un nuevo archivo.

3. En este editor introducimos la función mediante líneas de código. Ver figura 3.12.

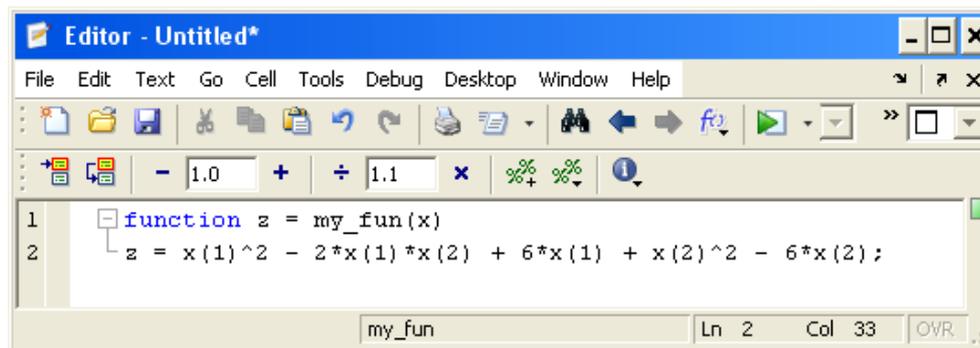


Figura 3.12: Editor de MATLAB.

4. Salvamos el fichero M-File en un directorio de MATLAB, como “Ejemplo”.

Tras realizar estos pasos abrimos la toolbox correspondiente a algoritmos genéticos, para ello, desde la pantalla principal de MATLAB, procedemos como se vio en la figura 3.4.

Start > Toolboxes > Genetic Algorithm and Direct Search Toolbox > Genetic Algorithm Tool.

Una vez en la ventana “Optimization Tool” de MATLAB, se escribe el nombre de la función que queremos optimizar en el espacio “Fitness función” y el numero de variables, en este caso dos, en el espacio “Number of variables”, como se muestra en la figura 3.13.

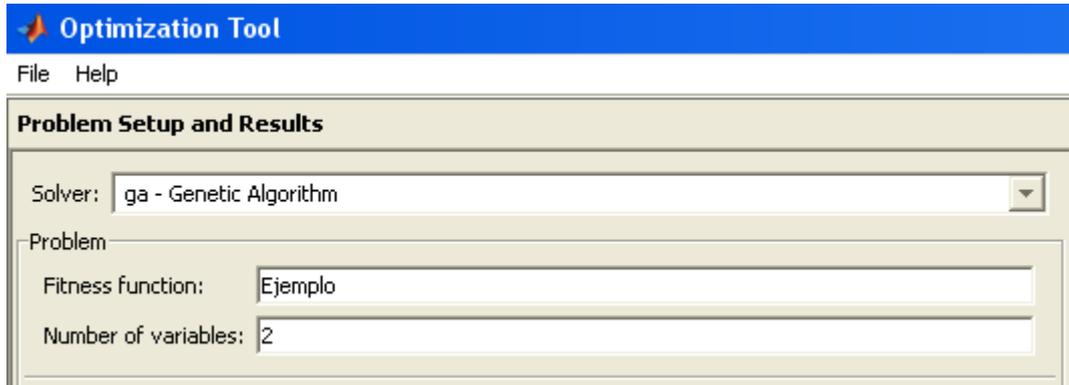


Figura 3.13: Introducción de la función.

Por otro lado, en “Plot functions” se define el gráfico para modelizar el proceso, para ello, seleccionaremos un “Plot interval” de valor unidad y un gráfico “Best Fitness”, como se muestra en la figura 3.14.

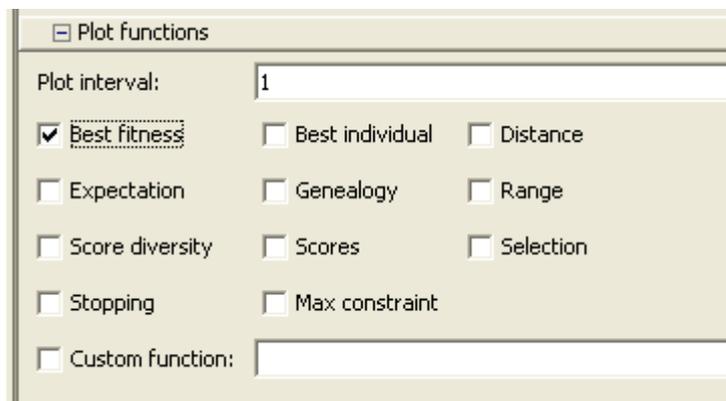


Figura 3.14: Definición del gráfico.

Del mismo modo, se debería de seleccionar las opciones correspondientes a: tipo de población, tamaño de la población, selección, cruce, et. Pero al tratarse de un simple ejemplo de implementación, se trabaja con una población real, sin codificación, por lo que no se han variado estos factores.

Una vez implementado el problema en la toolbox de MATLAB, procedemos a su ejecución, para ello pulsamos en “Run solver and view results” el botón “Start”, para que el programa comience a optimizar la función. Ver figura 3.15.

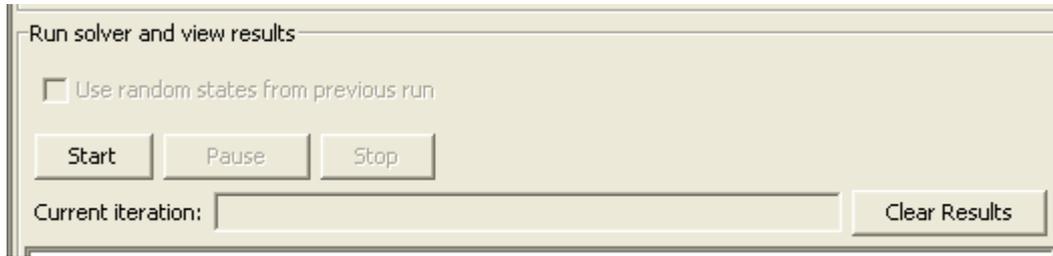


Figura 3.15: Resolución del problema.

Los resultados obtenidos pueden verse en “Run solver and view results”, donde se da el valor de aptitud del mejor individuo de la última generación, motivo de la parada del algoritmo. En este caso, esto ha sido debido a que no se ha realizado mejora del “Best fitness” en un número de generaciones definidas en “Stopping criteria”, que se ha dejado con su valor por defecto correspondiente a 50 generaciones.

En esta misma pantalla, podemos ver como en “Final point”, el programa también nos da el valor de las variables en el punto final de la iteración. Ver figura 3.16.

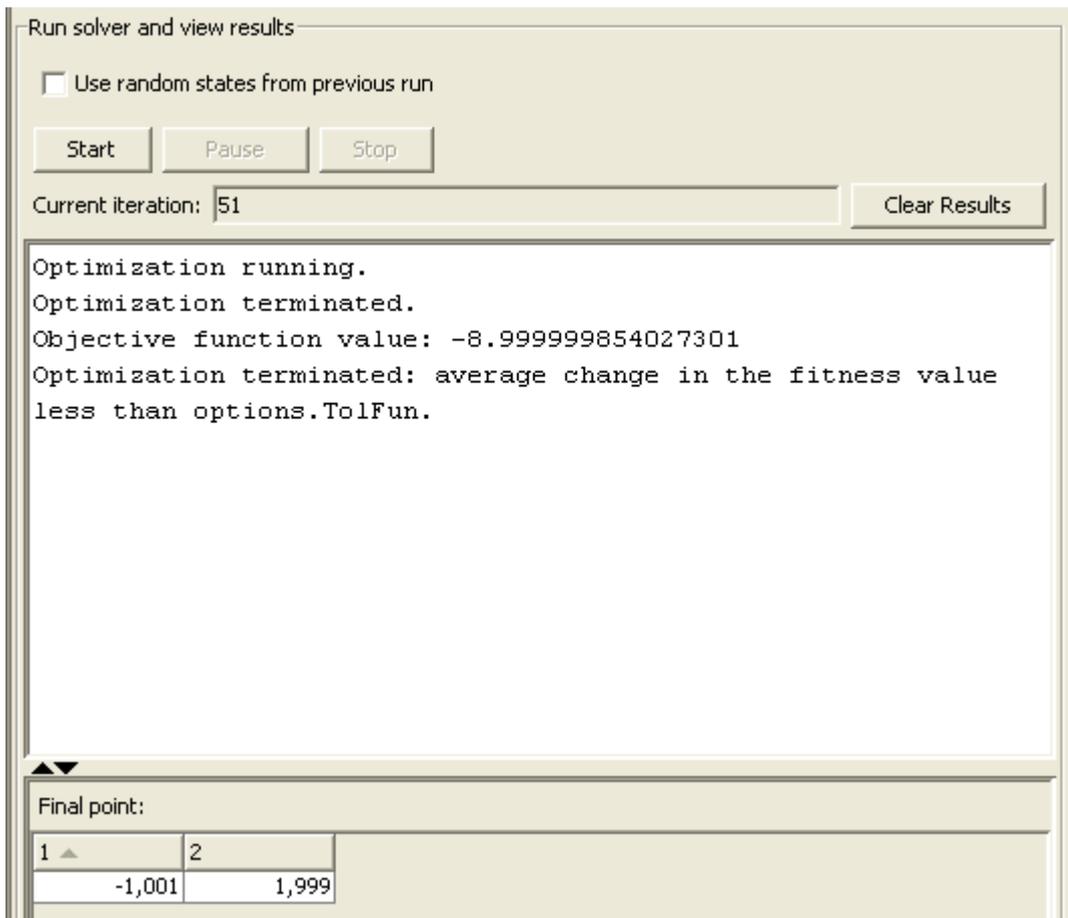


Figura 3.16: Puntos finales.

Por otro lado, como definimos el gráfico “Best fitness” podemos ver gráficamente como ha sido la evolución de la función, mediante el mejor valor de una generación “Best fitness” (Negro) y la media de toda la generación “Mean fitness” (Azul). Ver figura 3.17.

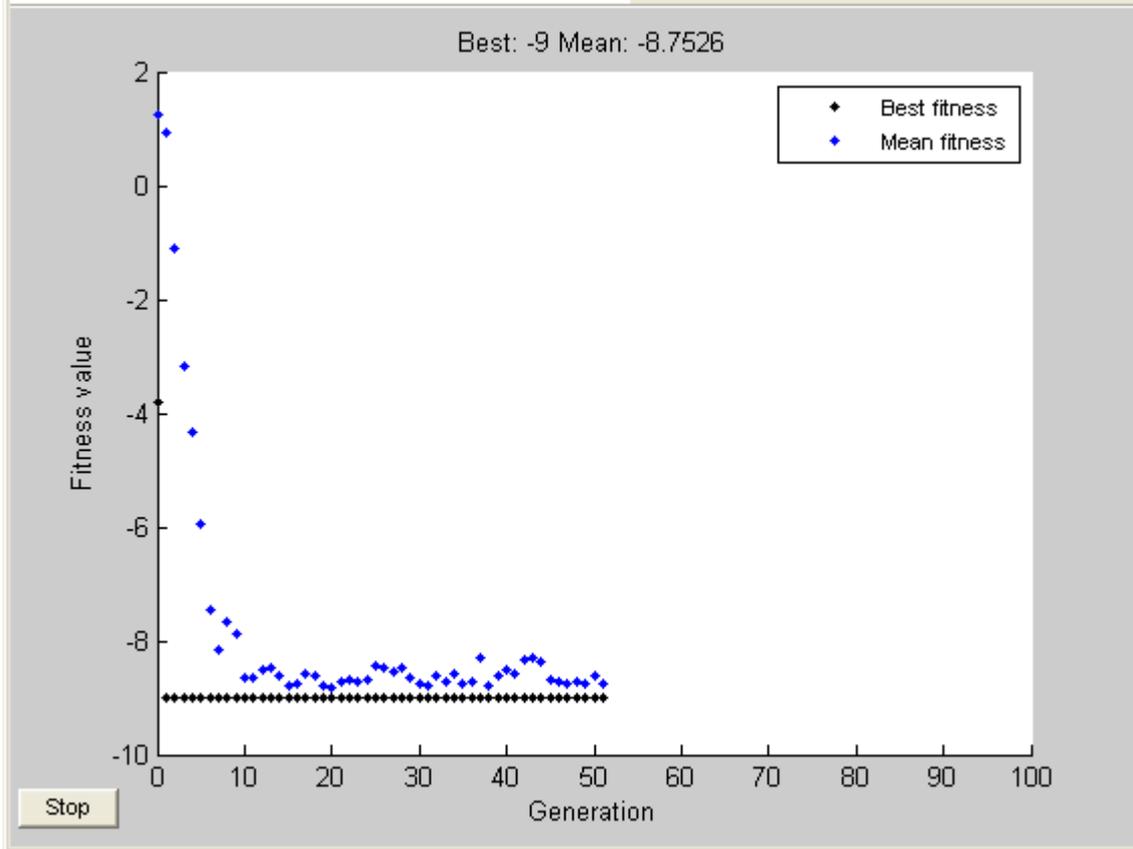


Figura 3.17: Gráfico “Best fitness”.

3.2.3. Ejemplo práctico de aplicación.

A continuación se desarrolla un ejemplo práctico de aplicación de algoritmos genéticos mediante la herramienta de cálculo Matlab.

3.2.3.1. Descripción del problema.

En este ejemplo, estudiaremos el diseño de tolerancias de un embrague de un sólo sentido que consta de, un soporte central (Hub), un anillo exterior (Cage) y cuatro rodillos (Roller). El contacto de los rodillos con el soporte central y el anillo exterior permite la transmisión de potencia del eje central a dicho anillo exterior, como se ve en la figura 3.18.

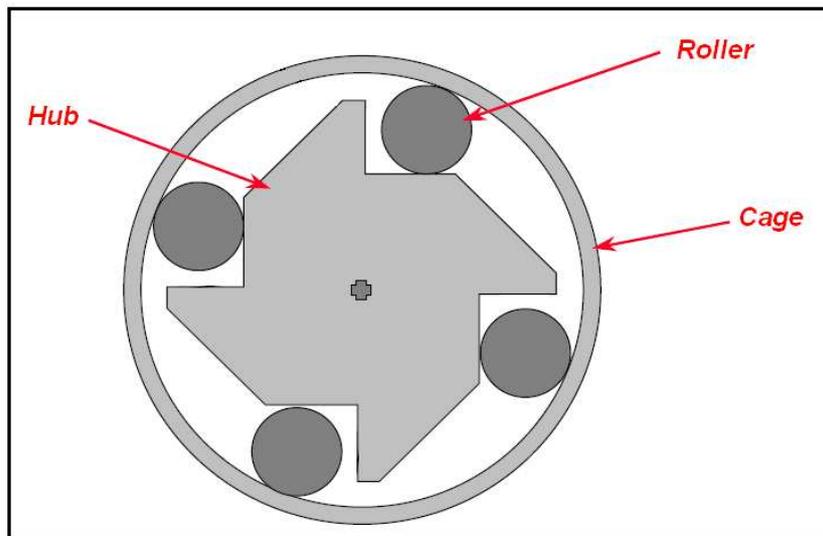


Figura 3.18: Esquema del problema. Ej. Embrague.

Para el correcto funcionamiento del embrague, será necesario conocer el ángulo de contacto y las dimensiones que afectan al ensamblado del conjunto.

Este problema fue descrito por A. Noorul en el artículo: "Tolerance desing optimization of machine elements using genetic algorithm". Noorul, H. et al (2005). Int. Journal Adv. Manuf. Technology, 25, 385-391, (Ver anexo [A.2]), y fue desarrollado en el proyecto fin de carrera: Aplicación de algoritmos genéticos a problemas de diseño en ingeniería mecánica utilizando Matlab por Javier Paz Fernández, en el curso 2006-2007.

3.2.3.2. Asignación de las variables y límites.

Como se ha mencionado anteriormente, y se puede ver en la figura 3.19, el embrague consta de tres partes diferenciadas de las que obtenemos tres variables independientes denominadas X_1, X_2, X_3 . Por otro lado, para su correcto funcionamiento, se define una variable Y correspondiente al ángulo de contacto, dependiente de las variables \mathbf{X} .

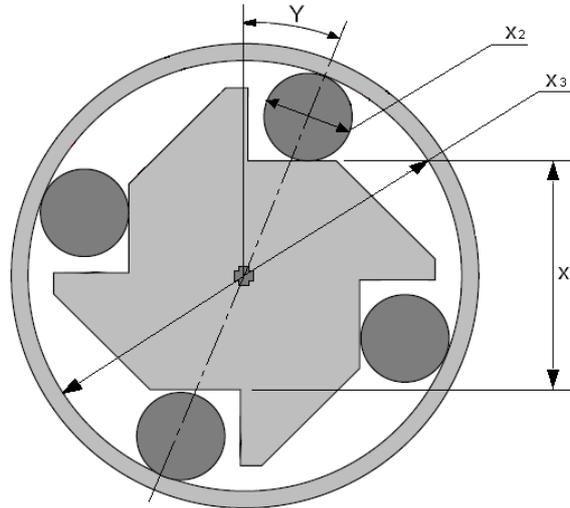


Figura 3.19: Asignación de variables. Ej. Embrague.

De manera que la variable Y se puede escribir como:

$$Y = f(x_1, x_2, x_3) = a \cdot \cos\left(\frac{x_1 + x_2}{x_3 - x_2}\right)$$

Donde "a" es una constante.

El valor de la variable Y deberá encontrarse dentro de unos límites de especificación tales que, LEI, LES .

$$Y = 0.122 \pm 0.035$$

Dada la dependencia de Y , el problema se basará en la optimización de las tolerancias de las variables \mathbf{X} , minimizando los costes.

Los límites de tolerancias LTI, LTS de las variables X_1, X_2, X_3 , serán:

$$\begin{aligned} X_1 &= 2.17706 \pm 0.012 \text{ in} \\ X_2 &= 0.90000 \pm 0.0005 \text{ in} \\ X_3 &= 4.00000 \pm 0.012 \text{ in} \end{aligned}$$

Siendo los valores mínimos para las tolerancias de $t_i = 0,0001 \text{ in}$

Los costes de fabricación se asignarán mediante las funciones:

$$M_1(t_1) = -0.731 + \frac{0.058}{t_1^{0.688}}$$

$$M_2(t_2) = -8.3884 + \frac{5.7807}{t_2^{0.0784}}$$

$$M_3(t_3) = -0.978 + \frac{0.0018}{t_3}$$

Donde t_1, t_2, t_3 son los valores de las tolerancias de las variables X , respectivamente.

El coste total de fabricación se puede expresar como:

$$M(t_1, t_2, t_3) = M_1(t_1) + M_2(t_2) + M_3(t_3)$$

La función del coste asociado a la pérdida de calidad se puede expresar como:

$$Q(t_1, t_2, t_3) = \sum_{k=1}^k \left(\frac{A}{T_k^2} \right) \cdot \sigma_k^2$$

Donde :

“A” es un coeficiente de pérdida de calidad medio en función de la variabilidad obtenida en Y .

“ T_Y ” es la tolerancia de la variable Y , que será función de las tolerancias de las variables X , es decir de las variables t_1, t_2, t_3 , y será calculada aplicando la ley de las varianzas.

Uniendo ambos costes, la función a minimizar será:

$$Y(t_1, t_2, t_3) = \sum_{i=1}^3 M_i(t_i) + Q(t_1, t_2, t_3)$$

$$Y(t_1, t_2, t_3) = \frac{0.058}{t_1^{0.688}} + \frac{23.1228}{t_2^{0.0784}} + \frac{0.0018}{t_3} + A(0.7029 t_1^2 + 362.811 t_2^2 + 90.7029 t_3^2) - 33.306$$

Las restricciones del problema serán los límites de las tolerancias t_1, t_2, t_3 , y la restricción para la tolerancia T_Y de la variable Y puede ser escrita en función de las tolerancias t_1, t_2, t_3 , aplicando la ley de varianzas y teniendo en cuenta que la tolerancia de Y está establecida como $\pm 0,035$.

$$107.1429 t_1 + 426.9714 t_2 + 106.3429 t_3 \leq 1$$

3.2.3.3. Planteamiento del problema para la aplicación de AG.

Para resolver el problema con algoritmos genéticos, será necesario:

- Buscar una adecuada codificación de las posibles soluciones del problema.

Dado que los valores de las variables son reales, se usará la expresión (3.2):

$$x = x_{\min} + bin2dec \left(\frac{x_{\max} - x_{\min}}{2^L - 1} \right)$$

Por lo tanto, será necesario definir el grado de precisión, los valores máximo y mínimo ya los da el artículo como límites de tolerancia para cada variable.

- Escribir una función objetivo que incluya las restricciones, ya que se usará el método de la penalización.

Codificación binaria.

Para saber la longitud de cada variable recordaremos la formula (3.1):

$$L = \log_2 \left(1 + \frac{x_{\max} - x_{\min}}{m} \right)$$

En el artículo se da una longitud de 19 bits a cada variable, pero para este caso, el programa no puede trabajar con más de 52 bits por lo que se ha decidido recalcular la longitud.

Los valores máximos y mínimo de cada variable se toman directamente del artículo en forma de límites de tolerancia y se utilizará una precisión de 0.0001 para todas las variables.

Se han tomado como valores máximos y mínimos 0.012 y 0.0001 respectivamente, los cuales han sido sacados de los límites de tolerancia t_1, t_2, t_3 . La precisión que se ha usado es de 0.0001, por lo que sustituyendo tenemos:

$$L = \log_2 \left(1 + \frac{x_{\max} - x_{\min}}{m} \right) = \log_2 \left(1 + \frac{0.012 - 0.0001}{0.0001} \right) = 7$$

Es decir, $2^7 = 128 > 120$, por lo que esta longitud puede representar las variables. Por lo tanto, cada variable será de 7 bits y la longitud total de la cadena será de 21 bits.

3.2.3.4. Construcción de la función objetivo.

Se usará la función objetivo:

$$Y = \frac{0.058}{t_1^{0.688}} + \frac{23.1228}{t_2^{0.0784}} + \frac{0.0018}{t_3} + A(0.7029 t_1^2 + 362.811 t_2^2 + 90.7029 t_3^2) - 33.306$$

pero se le agrega una penalización correspondiente a la restricción:

$$107.1429 t_1 + 426.9714 t_2 + 106.3429 t_3 \leq 1$$

3.2.3.5. Programación en MATLAB.

La función creada en el archivo *.m con codificación binaria será:

funcion y=fortini(x)

```
%a=coeficiente estimado de pérdida de calidad.
%y=coste total de fabricación y pérdida de la calidad.
%t1=single side tolerance value para dimensión 1.
%t2=single side tolerance value para dimensión 2.
%t3=single side tolerance value para dimensión 3.
```

```
A=52;
long=7;
```

```
t1min=0.0001;
t2min=0.0001;
t3min=0.0001;
t1max=0.012;
t2max=0.0005;
t3max=0.012;
```

```
xmin=[t1min t2min t3min];
xmax=[t1max t2max t3max];
```

```
b=x(1:long);
c=x(long+1:2*long);
d=x(2*long+1:3*long);
```

```
x1=num2str(b);
x2=num2str(c);
x3=num2str(d);
```

```
t1=xmin(1)+bin2dec(x1)*((xmax(1)-xmin(1))/(2^long-1));
t2=xmin(2)+bin2dec(x2)*((xmax(2)-xmin(2))/(2^long-1));
t3=xmin(3)+bin2dec(x3)*((xmax(3)-xmin(3))/(2^long-1));
```

```

r=107.1429*t1+426.9714*t2+106.3429*t3;

p=0;

if r>1
    p=100000;
end

y=(0.058/(t1^0.688))+(23.1228/(t2^0.0784))+(0.0018/(t3))+A*(90.7029*(t1^2)+362.811*(t2^2)+9
0.7029*(t3^2))-33.306+p;

```

“r” es la restricción que penaliza la función objetivo si no se cumple.

En este caso como lo que se pide es minimizar, la función objetivo no se pone negativo, ya que el programa por defecto lo que hace es minimizar al máximo la función.

El problema se resolverá para diferentes valores del coeficiente de pérdida de carga “A” para comparar los resultados con los del artículo. Por lo tanto este valor se modificará en el programa, en este caso de ejemplo toma el valor A=52.

3.2.3.6. Parámetros de AGs usados.

Parámetros del algoritmo genético.

Los parámetros del algoritmo genético son:

Population size = 140.

Cros-over probability = 0.6

Mutation probability = 0.03.

Estos datos se sacan directamente de los datos en el artículo.

Las otras opciones usadas en AG son las siguientes:

1. Fitness función = @fortini. Que es el nombre de la función.
2. Number of variables = 21. Se tiene tres variables que da el problema y cada una con 7bits.
3. Plots = Best fitness. Para que se vea cómo evoluciona la función objetivo.
4. Population type = Bit string. Se va a trabajar con números binarios en todos los problemas, por lo que esta opción será la misma en todos.
5. Population size = 140. Dato del problema.
6. Selection o uniform: Se usa la que obtenga mejores resultados, aunque muchas veces no varíe prácticamente nada de una a otra.
7. Reproduction: crossover fraction = 0.6. Dato que se da en el problema.

8. Mutación función: uniform. Rate = 0.03. Valor de la probabilidad que tiene cada bit de mutar, esta dado por el problema.

9. Crossover: single point.

10. Stopping criteria: generations: 200, stall generation: 100, stall time limit: 200.

11. Run solver.

3.2.3.7. Conclusión.

En la tabla 3.9 podemos ver los resultados dados en el artículo “Tolerance desing optimization of machine elements using genetic algorithm”. Noorul, H. et al (2005), para diferentes valores de A y compararlos con los resultados de la tabla 3.10, obtenidos mediante la aplicación de algoritmos genéticos con Matlab, Genetic Algorithm and Direct Search Toolbox con codificación binaria.

Tabla 3.9: Resultados dados en el artículo. Ej. Embrague.

A	t ₁	t ₂	t ₃	F. Objetivo
0	0,00495	0,0005	0,00239	11,640
1	0,00486	0,0005	0,00248	11,611
52	0,00444	0,0005	0,00225	11,793
100	0,00481	0,0005	0,00210	11,918
300	0,00467	0,0005	0,00267	12,466
520	0,00425	0,0005	0,00267	13,047

Tabla 3.10: Resultados obtenidos con Matlab. Ej. Embrague.

A	t ₁	t ₂	t ₃	F. Objetivo
0	0,0053	0,0005	0,0020	11,6871
1	0,0051	0,0005	0,0023	11,6560
52	0,0050	0,0005	0,0023	11,7976
100	0,0045	0,0005	0,0028	11,9450
300	0,0046	0,0005	0,0027	12,4726
520	0,0042	0,0005	0,0026	13,0481

Tras realizar varias ejecuciones del problema se puede concluir con que los resultados obtenidos para diferentes valores del coeficiente estimado de pérdida de carga, A, mediante la aplicación de algoritmos genéticos en Matlab, son bastante aproximados a los obtenidos en el artículo propuesto, por lo tanto, utilizaremos esta programación para la resolución de la metodología en el tema IV.

4. CAPÍTULO CUARTO: METODOLOGÍA PARA LA ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES A UN CONJUNTO DE VARIABLES DEPENDIENTES

En este capítulo se describe detalladamente la metodología propuesta. En un primer apartado se define el problema de la asignación de tolerancias y valores nominales a un conjunto de variables dependientes. En un segundo apartado se describe paso a paso la metodología propuesta. En el capítulo siguiente se ilustrará la metodología propuesta con varios ejemplos y aplicando como herramienta de optimización los algoritmos genéticos.

4.1. DEFINICIÓN DEL PROBLEMA.

El problema que se va a tratar fue definido previamente en el apartado 2.2.2 del capítulo II. Se trata de asignar las tolerancias y los valores nominales a un conjunto de variables correladas, de manera que se maximicen las tolerancias y se obtenga una proporción de piezas defectuosas menor que cierto valor α .

Para facilitar la lectura y comprensión de la metodología que se propone, se define aquí nuevamente el problema.

Consideremos que una serie de m variables $\mathbf{Y} = (Y_1, Y_2, \dots, Y_m)$, son unas características de las que depende el correcto funcionamiento de una pieza o conjunto mecánico. Podemos considerar entonces que esta pieza o conjunto funcionará correctamente si estas variables \mathbf{Y} , son conformes, es decir, están dentro de unos límites de especificación marcados, esto es:

$$Y_j \in [LEI_j, LES_j], j = 1, \dots, m$$

Siendo, LEI_j y LES_j , los límites de especificación inferior y superior, respectivamente, de cada una de las variables \mathbf{Y} .

Se asume, que bajo condiciones óptimas de fabricación, la proporción de piezas defectuosas, denotada por p , no debe ser mayor que un pequeño valor α , es decir:

$$p \equiv P(\mathbf{Y} \notin S) \leq \alpha \quad (4.1)$$

donde S es la región rectangular definida por las especificaciones de la pieza, por lo que:

$$S = \{ \mathbf{Y} \in R^m : (LES_j \leq Y_j \leq LEI_j), j = 1, \dots, m \} \quad (4.2)$$

Por otro lado, las variables \mathbf{Y} dependen de un conjunto de k variables correladas o dependientes $\mathbf{X} = (X_1, X_2, \dots, X_k)$, es decir:

$$\mathbf{Y} = f[X_1, X_2, \dots, X_k]$$

Estas variables \mathbf{X} son características de la pieza y están correladas debido al proceso de fabricación de las piezas.

En el caso de las variables \mathbf{Y} , dado que dependen de las mismas variables \mathbf{X} se considerará también que existe correlación entre ellas.

El problema consiste en encontrar las tolerancias óptimas $\mathbf{t} = (t_1, t_2, \dots, t_k)$ y los valores nominales óptimos $\mathbf{VN} = (VN_1, VN_2, \dots, VN_k)$, de cada una de las k variables \mathbf{X} . Los valores óptimos serán aquellos que permitan maximizar las tolerancias cumpliendo la restricción de la ecuación 4.1, es decir, obteniendo una proporción de defectuosos menor que cierto valor α .

Dadas las tolerancias y valores nominales óptimos, los límites de tolerancia de las variables \mathbf{X} serán los siguientes:

$$\begin{aligned} LTI_i &= VN_i - t_i \\ LTS_i &= VN_i + t_i, \quad i = 1, 2, \dots, k \end{aligned}$$

Para encontrar los valores nominales óptimos, se restringirán sus valores a un intervalo determinado, del tipo:

$$VN_i \in [VNI_i, VNS_i], \quad i = 1, 2, \dots, k.$$

siendo VNI_i y VNS_i el valor nominal inferior y valor nominal superior, respectivamente.

Del mismo modo, se establecerán unas tolerancias mínimas T_{i_min} , a cada variable \mathbf{X} , para evitar que las tolerancias tomen valor igual a cero o próximo a cero.

Por otro lado, se asume igual coste de fabricación y calidad para todas las variables \mathbf{X} . Es decir, ninguna de ellas tiene mayor coste de calidad ni existe distinto coste para obtener la misma tolerancia en cualquiera de las variables.

Luego, el problema quedaría planteado como:

$$\begin{aligned} \max \quad \mathbf{t} &= (t_1, t_2, \dots, t_k) = \max \sum_{i=1}^k t_i \\ \text{sujeto a} \\ p &\leq \alpha \\ VN_i &\in [VNI_i, VNS_i], \quad i = 1, 2, \dots, k. \\ t_i &\geq T_{i_min} \end{aligned} \tag{4.3}$$

Para resolver este problema se asumirán muchas de las condiciones vistas en las metodologías propuestas para el caso de independencia de \mathbf{X} , descritas es el apartado 2.3.

Se asume la normalidad multivariante de las variables \mathbf{X} , por lo que se puede decir que:

$$\mathbf{X} \approx N_K(\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X^0)$$

donde $\boldsymbol{\mu}_X$ es el vector de medias y $\boldsymbol{\Sigma}_X^0$ es la matriz de covarianzas, ambos dependientes del proceso de fabricación.

Se asume también una relación lineal entre las variables \mathbf{X} e \mathbf{Y} , por lo que cada variable Y_j , puede ser expresada como:

$$Y_j = a_{0j} + a_{ij}X_i + \dots + a_{kj}X_k$$

Cuando la relación lineal sea sólo aproximadamente cierta, se podrá aplicar un desarrollo de Taylor, en este caso los componentes a_{ij} , podrán ser definidos como:

$$a_{ij} = \frac{\delta Y_j}{\delta X_{ij}}$$

y el coeficiente a_{0j} se obtiene a partir del desarrollo de Taylor.

Bajo la suposición de linealidad, se tiene que, $\mathbf{Y} \approx N_m(\boldsymbol{\eta}_y; \boldsymbol{\Sigma}_y^0)$, por lo que el vector de medias verifica que:

$$\boldsymbol{\eta}_y = \mathbf{a} + \mathbf{A}\boldsymbol{\mu}_x' \quad (4.4)$$

donde $\mathbf{a} = (a_{01}, a_{02}, \dots, a_{0j})$, y \mathbf{A} es la matriz de relación.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{k1} \\ a_{12} & a_{22} & \cdots & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{km} \end{bmatrix}$$

y la matriz de covarianzas $\boldsymbol{\Sigma}_y^0$, verifica que:

$$\boldsymbol{\Sigma}_y^0 = \mathbf{A}\boldsymbol{\Sigma}_x^0\mathbf{A}' \quad (4.5)$$

Al igual que en el caso de independencia de \mathbf{X} , donde la maximización de t_i dependía de la varianza de las variables, ahora bajo dependencia de \mathbf{X} el valor de las tolerancias t_i dependerá de la matriz de covarianzas Σ_x . En este caso, sin embargo, los valores nominales alrededor de los cuales se establecerán las tolerancias son también variables de diseño. Luego, asumiendo que el proceso estará centrado en los valores nominales, el problema de asignación de tolerancias dependerá también de los valores del vector de medias μ_x . De este modo, obtendremos una generalización mayor donde la maximización de las tolerancias implicará Σ_x y μ_x .

4.2. METODOLOGÍA PROPUESTA PARA LA ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES.

La metodología que se propone en este proyecto está dividida en dos etapas que se describirán a continuación.

El problema que se plantea en esta metodología consiste en asignar las tolerancias y los valores nominales de \mathbf{X} , de manera que se maximicen las tolerancias obteniendo una proporción de defectuosos $p \leq \alpha$. Por tanto, es necesario establecer un procedimiento para calcular dicha proporción p en cada iteración del proceso de optimización.

Puesto que este cálculo de p se realizará para cada iteración del proceso de optimización y puede requerir complicadas integraciones numéricas o simulaciones de Monte Carlo, en esta metodología se propone establecer dos procedimientos para la evaluación de dicha restricción, lo que nos llevará a dividir la metodología en dos etapas.

En la Etapa I, se realizará un procedimiento simplificado para obtener una evaluación rápida de $p \leq \alpha$ sin necesidad de calcular el valor exacto de p . Como resultado de esta etapa obtendremos unos valores nominales y unas tolerancias de \mathbf{X} . Estas tolerancias, serán ajustadas en la etapa II, en la que se hará un cálculo más preciso de la proporción p .

En la Etapa II se calculará la proporción p mediante un procedimiento basado en simulaciones de Monte Carlo. Dado que este procedimiento es más costoso computacionalmente se realizará como último paso en la resolución del problema. Como resultado de esta etapa obtendremos las tolerancias óptimas de \mathbf{X} .

Antes de describir detalladamente los pasos a seguir en la metodología, se explicará a grandes rasgos en qué consiste. Partiremos de una matriz de covarianzas Σ_x y un vector de medias μ_x , obtenidos con datos del proceso de fabricación trabajando en condiciones normales. Dada la dependencia establecida entre las variables \mathbf{X} e \mathbf{Y} , calcularemos los valores de las variables \mathbf{Y} mediante (4.4) y (4.5). Una vez obtenidos los valores η_Y y Σ_Y se calculará la proporción de artículos defectuosos que produciría esta distribución.

En la Etapa I, este cálculo se hará utilizando la proyección de una región elíptica a una rectangular, explicada en el capítulo II. Si la proporción de defectuosos no es la que buscamos, entonces se modificará la matriz de covarianzas y el vector de medias de \mathbf{X} , obteniendo unos nuevos valores, Σ_x^* y μ_x^* , y unos nuevos valores η_Y^* y Σ_Y^* .

Este proceso será repetido tantas veces como sea necesario hasta cumplir con la proporción deseada, es decir, $p \leq \alpha$.

Una vez terminada esta Etapa I, se realizará un proceso basado en simulaciones de Monte Carlo donde se ajustará de una forma más precisa la proporción de no conformidad deseada $p \leq \alpha$. Para ello, partiremos de los datos Σ_x^* y μ_x^* , η_y^* y Σ_y^* obtenidos en la última iteración de la Etapa I. Se simularán observaciones de una distribución normal multivariante de parámetros η_y^* y Σ_y^* y con estas observaciones se calculará la proporción que están fuera de las especificaciones, es decir la proporción de defectuosos.

A continuación se explica detalladamente cada paso de la metodología propuesta.

4.2.1. Etapa I.

Para el desarrollo de esta primera etapa es necesario resolver los siguientes aspectos:

- Procedimiento para calcular el porcentaje de defectuosos.
- Procedimiento para modificar la matriz de covarianzas Σ_x .
- Procedimiento para modificar el vector de medias μ_x .

Procedimiento para calcular el porcentaje de defectuosos

En esta primera etapa se calculará la proporción de defectuosos p de manera aproximada. Para ello se aplicará la metodología descrita en el apartado 2.1.11 del capítulo II, que permite obtener una región de confianza rectangular a partir de un hiperelipsoide de variabilidad $1-\alpha$. En este caso el hiperelipsoide corresponderá a la distribución de las variables Y , es decir al definido por una matriz de covarianzas Σ_y y un vector de medias η_y .

Como se puede observar en la figura 4.1, para el caso de dimensión $k=2$, α es la proporción de valores que se encuentran fuera de la elipse de variabilidad $1-\alpha$. Sin embargo, el rectángulo proyectado tangente a la elipse abarca, claramente, una región más grande que la propia elipse por lo que la proporción fuera de este rectángulo será menor que α .

Lo que se buscará es que este rectángulo sea igual a un rectángulo formado por las especificaciones de Y . Pero, incluso cuando estos dos rectángulos sean iguales, se tendrá una proporción de defectuosos $p < \alpha$. Luego, en esta primera etapa obtendremos una solución conservadora que podrá ser ajustada en la segunda etapa.

Es importante mencionar que las especificaciones del problema se consideran, unilaterales, $Y \leq 127.3$ o bilaterales, $126.7 \leq Y \leq 127.3$, dependiendo de los requerimientos establecidos por el problema que se vaya a realizar

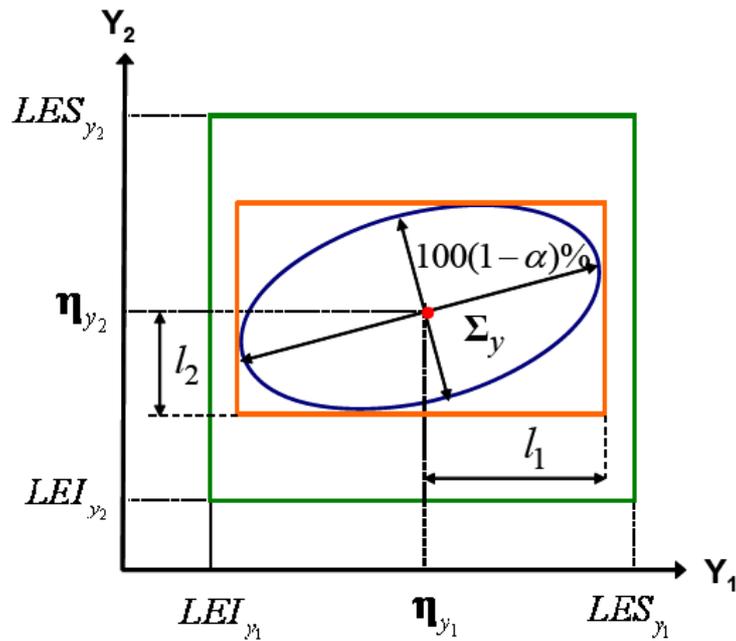


Figura 4.1: Etapa I, aproximación $p < \alpha$.

Procedimiento para modificar la matriz de covarianzas Σ_x .

Como vimos en el caso de independencia de \mathbf{X} , donde las variables a optimizar eran las desviaciones típicas σ_i , las cuales eran tratadas como independientes unas de otras, ahora la dependencia de \mathbf{X} proviene de la matriz de covarianzas Σ_x , que se convertirá en una de las variables de optimización.

Para cada iteración del proceso de optimización será necesario cambiar Σ_x , lo que implicará un nuevo conjunto de límites de tolerancia y una nueva Σ_y que llevará a una nueva proporción de defectuosos.

Para modificar la matriz de covarianzas Σ_x , se aplicará la metodología propuesta por González y Sánchez (2008) basada en el análisis de componentes principales (ACP) descrito en el apartado 2.1.12 del capítulo II. Los autores justifican la aplicación de esta metodología del siguiente modo.

Las variables \mathbf{X} no son independientes pero pueden ser consideradas como una combinación de factores independientes $\mathbf{Z} = (Z_1, Z_2, \dots, Z_K)'$ denominados factores latentes. Estos factores independientes o latentes estarán relacionados directamente con el proceso de fabricación y pueden ser interpretados como la fuente primaria de la estructura dependiente de \mathbf{X} . Por tanto, los cambios en Σ_x serán provocados por cambios en la varianza de estos factores independientes \mathbf{Z} , siendo estas modificaciones compatibles con las características del proceso de fabricación. La obtención de estos factores independientes \mathbf{Z} se hará mediante el análisis de componentes principales.

Como vimos en el capítulo II el ACP permite pasar de un conjunto de variables dependientes a un conjunto de variables independientes mediante la descomposición de la matriz de covarianzas. Así, suponiendo que las variables tienen unas covarianzas Σ_x^0 , y aplicando ACP, se obtendría:

$$\Sigma_x^0 = \mathbf{C}\mathbf{D}^0\mathbf{C}' \quad (4.6)$$

donde \mathbf{C} es la matriz de autovectores de dimensión $(k \times k)$ y \mathbf{D}^0 , es la matriz diagonal de autovalores $(\lambda_1^0, \lambda_2^0, \dots, \lambda_k^0)$ que representan las varianzas de los factores independientes \mathbf{Z} . Luego, modificando los autovalores modificaremos la variabilidad de los factores independientes y con ello modificaremos la matriz de covarianzas de \mathbf{X} .

Geoméricamente, los autovectores representaran las direcciones de los ejes del hiperelipsoide definido por Σ_x^0 y los autovalores el tamaño de estos, por lo tanto, un cambio en los autovalores producirá un cambio de longitud de los ejes y un cambio en los autovectores un cambio en la dirección de estos.

La metodología empleada en este proyecto respeta la relación de dependencia entre variables \mathbf{X} , por tanto, se variará la magnitud de las varianzas y las covarianzas pero se mantendrá la misma estructura. Esto se conseguirá modificando sólo los autovalores y manteniendo constante los autovectores.

Es importante mencionar aquí que la matriz de covarianzas Σ_x^0 depende del proceso de fabricación de la pieza. Debe ser estimada con una muestra suficientemente grande de datos obtenida del proceso de fabricación funcionando en condiciones normales, sin fallos y con una variabilidad normal, es decir, estando el proceso bajo control.

Una vez obtenidos los autovalores $(\lambda_1^0, \lambda_2^0, \dots, \lambda_k^0)$ que representan las varianzas de \mathbf{Z} , modificaremos su valor multiplicándolos por una serie de k coeficientes b_i , siendo $i = 1, 2, \dots, k$, obteniendo los nuevos valores $(\lambda_1^*, \lambda_2^*, \dots, \lambda_k^*)$, esto es:

$$\begin{aligned} \lambda_1^* &= \lambda_1^0 \cdot b_1 \\ \lambda_2^* &= \lambda_2^0 \cdot b_2 \\ &\vdots \end{aligned} \quad (4.7)$$

donde b_i es un coeficiente positivo.

Dado que los autovalores representan geoméricamente el tamaño de los ejes del hiperelipsoide, el uso de diferentes valores b_i implicará el cambio de la forma del hiperelipsoide, mientras que, la aplicación del mismo valor b_i a todos los autovalores provocara que el hiperelipsoide aumente o disminuya su tamaño respetando su forma inicial. En ambos casos, la dirección original de los ejes no cambia.

En esta primera etapa de la metodología se usarán distintos valores de b_i de manera que la modificación de Σ_x^0 es más general.

Una vez obtenidos los nuevos valores $(\lambda_1^*, \lambda_2^*, \dots, \lambda_k^*)$ calculamos la nueva matriz de covarianzas Σ_x^* siendo:

$$\Sigma_x^* = \mathbf{CD}^*\mathbf{C}' \quad (4.8)$$

donde \mathbf{D}^* , es la nueva matriz diagonal de autovalores $(\lambda_1^*, \lambda_2^*, \dots, \lambda_k^*)$.

Mediante este procedimiento podemos ver que los cambios producidos en la matriz de covarianzas inicial Σ_x^0 dependen del valor de los k coeficientes b_i y por tanto la optimización de dicha matriz puede ser tratada como la optimización de estos coeficientes independientes, convirtiendo este procedimiento en un problema de optimización similar al caso bajo independencia de \mathbf{X} , visto en el capítulo II, donde las variables a optimizar eran las k desviaciones estándar independientes.

Una vez calculada la nueva matriz Σ_x^* es posible calcular la nueva matriz de covarianzas de \mathbf{Y} como $\Sigma_y^* = \mathbf{A}\Sigma_x^*\mathbf{A}'$. Esta matriz Σ_y^* será utilizada en cada iteración para calcular la proporción de defectuosos según el procedimiento descrito arriba.

Procedimiento para modificar el vector de medias μ_x .

La maximización de tolerancias implica también seleccionar los valores nominales de las variables \mathbf{X} . Estos valores nominales se suponen iguales a las medias de las variables y por tanto, modificar sus valores implica modificar los valores del vector de medias μ_x^0 .

Mediante el análisis de componentes principales, la serie de k variables dependientes \mathbf{X} , que definen las medidas del conjunto mecánico pueden considerarse, como vimos anteriormente, como una combinación lineal de k factores independientes. Si consideramos que tenemos una matriz de datos de las variables \mathbf{X} , de dimensión $(n \times p)$, que denotaremos como $\bar{\mathbf{X}}$, entonces los factores independientes pueden ser calculados como:

$$\mathbf{Z} = \bar{\mathbf{X}}\mathbf{C} \quad (4.9)$$

donde $\bar{\mathbf{X}}$, es la matriz de dimensión $(n \times p)$ que podemos definir como:

$$\bar{\mathbf{X}} = (\mathbf{X} - \mathbf{1}\boldsymbol{\mu}_x^{0'}) \quad (4.10)$$

Siendo, $\boldsymbol{\mu}_x^{0'}$ la transpuesta del vector de medias y $\mathbf{1}$ un vector $(n \times 1)$.

El desajuste o cambio del vector de medias de \mathbf{Z} , se realizará sumando a cada variable un valor g_i , siendo $i=1,2,\dots,k$, que podrá ser tanto positivo como negativo.

$$\mathbf{Z}^* = \mathbf{Z} + \mathbf{1}\mathbf{G}'$$

donde, \mathbf{G}' es la transpuesta del vector de valores g_i , es decir, $\mathbf{G} = (g_1, g_2, \dots, g_k)'$ y $\mathbf{1}$ un vector de unos $(n \times 1)$.

El valor esperado, o media, de las nuevas variables \mathbf{Z}^* , será:

$$E(\mathbf{Z}^*) = E(\mathbf{Z}) + E(\mathbf{1}\mathbf{G}')$$

Como $E(\mathbf{Z}) = 0$, tenemos que el valor esperado o media de \mathbf{Z}^* es:

$$\begin{aligned} E(\mathbf{Z}^*) &= 0 + E(\mathbf{1}\mathbf{G}') \\ E(\mathbf{Z}^*) &= \mathbf{G} \end{aligned}$$

El cambio producido en el valor de la media en las variables \mathbf{Z} se trasladará a las variables \mathbf{X} , es decir, partiendo de las ecuaciones (4.9) y (4.10), tenemos que:

$$\left. \begin{aligned} \mathbf{Z} &= \bar{\mathbf{X}}\mathbf{C} \Rightarrow \bar{\mathbf{X}} = \mathbf{Z}\mathbf{C}' \\ \bar{\mathbf{X}} &= (\mathbf{X} - \mathbf{1}\boldsymbol{\mu}_x^{0'}) \end{aligned} \right\} \begin{aligned} \mathbf{Z}\mathbf{C}' &= (\mathbf{X} - \mathbf{1}\boldsymbol{\mu}_x^{0'}) \\ \mathbf{X} &= \mathbf{Z}\mathbf{C}' + \mathbf{1}\boldsymbol{\mu}_x^{0'} \end{aligned}$$

Por lo tanto, para obtener un nuevo vector de medias de \mathbf{X} se debe usar la siguiente expresión:

$$\mathbf{X}^* = \mathbf{Z}^*\mathbf{C}' + \mathbf{1}\boldsymbol{\mu}_x^{0'}$$

$$\boxed{E(\mathbf{X}^*) = \boldsymbol{\mu}_x^* = \mathbf{C}\mathbf{G} + \boldsymbol{\mu}_x^{0'}} \quad (4.11)$$

Una vez obtenido este nuevo vector de medias $\boldsymbol{\mu}_x^*$, que representa un nuevo vector de valores nominales de \mathbf{X} , la hiperelipse ya no está centrada en $\boldsymbol{\mu}_x^0$, si no que se ha desplazado y está centrada en $\boldsymbol{\mu}_x^*$.

Por tanto, en el problema de maximización de tolerancias se tendrán como variables de optimización, además de los k coeficientes b_i , los k coeficientes g_i .

A continuación se resumen brevemente los pasos a seguir en esta Etapa I:

1. Inicializamos la iteración mediante $r = 0$, Estimamos la matriz de covarianzas y el vector de medias iniciales $\boldsymbol{\Sigma}_x^0$ y $\boldsymbol{\mu}_x^0$, con datos obtenidos del proceso de fabricación trabajando bajo control.
2. Mediante el ACP, usando $\boldsymbol{\Sigma}_x^0$ en (4.6), obtenemos la matriz de autovectores \mathbf{C} y la matriz diagonal inicial \mathbf{D}^0 , de autovalores $\lambda_1^0, \lambda_2^0, \dots, \lambda_k^0$.
3. Obtenemos el vector de medias de η_y mediante el reemplazo de $\boldsymbol{\mu}_x^0$ en (4.4).
4. Obtenemos la matriz de covarianzas $\boldsymbol{\Sigma}_y^0$ mediante el reemplazo de $\boldsymbol{\Sigma}_x^0$ en (4.5).
5. Obtenemos las longitudes l_j y calculamos sus límites $[LI_j, LS_j]$, mediante Nickerson (1994).
6. Si se cumple, $p \approx \alpha$, es decir, $[LI_j, LS_j] = [LEI_j, LEI_j]$; $j = 1, 2, \dots, m$, pasamos a la etapa II. Por el contrario continuamos con el paso 7.
7. Poner $r = r + 1$. Generamos una serie de k coeficientes b_i y, aplicando el procedimiento visto en esta sección, obtenemos una nueva matriz de covarianzas $\boldsymbol{\Sigma}_x^*$ mediante (4.8). Seguidamente, generamos un vector \mathbf{G} y obtenemos un nuevo vector de medias $\boldsymbol{\mu}_x^*$, mediante (4.11). Una vez obtenidas $\boldsymbol{\Sigma}_x^*$ y $\boldsymbol{\mu}_x^*$, volvemos al paso 3 para continuar con la iteración.

4.2.2. Etapa II.

En esta segunda etapa se ajustarán los resultados obtenidos en la primera etapa, calculando de manera más precisa la proporción de defectuosos. En este caso el cálculo de esta proporción se hará mediante simulaciones de Monte Carlo.

El método de Monte Carlo es un método estadístico numérico usado para aproximar expresiones matemáticas complejas y costosas de evaluar con exactitud. Se llamó así en referencia al Casino de Monte Carlo (Principado de Mónaco) por ser “la capital del juego de azar”, al ser la ruleta un generador simple de números aleatorios. El nombre y el desarrollo sistemático de los métodos de Monte Carlo datan aproximadamente de 1944 y se mejoraron enormemente con el desarrollo de la computadora. Este método numérico permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias. La importancia actual del método Monte Carlo se basa en la existencia de problemas que tienen difícil solución por métodos exclusivamente analíticos o numéricos, pero que dependen de factores aleatorios o se pueden asociar a un modelo probabilístico artificial (resolución de integrales de muchas variables, minimización de funciones, etc.). Gracias al avance en diseño de los ordenadores, los cálculos necesarios para la aplicación del método de Monte Carlo, que en otro tiempo hubieran sido inconcebibles, hoy en día se presentan como asequibles para la resolución de ciertos problemas.

En esta segunda etapa se partirá de la matriz de covarianzas y el vector de medias obtenidos en la última iteración de la Etapa I y se modificará únicamente la matriz de covarianzas considerando fijo el valor del vector de medias obtenido en la etapa I.

Dado que el coste computacional de las simulaciones de Monte Carlo es elevado, en esta etapa la modificación de la matriz de covarianzas se hará utilizando el mismo factor b en la ecuación (4.7). Esto significa que la longitud de los ejes de la elipse cambia de forma proporcional, es decir, el hiperelipsoide mantiene la misma forma (obtenida en la Etapa I) pero aumenta o disminuye de tamaño. Por tanto, si denotamos la matriz de covarianzas de \mathbf{X} , obtenida en la etapa I, como $\Sigma_x^{(I)}$, en cada iteración se obtendrá una nueva matriz como:

$$\Sigma_x^{(II)} = b\Sigma_x^{(I)} \quad (4.12)$$

Luego, en esta etapa, la variable a optimizar es el factor b .

Continuando con la metodología, una vez determinada la nueva matriz de covarianzas $\Sigma_x^{(II)}$, se usarán simulaciones de Monte Carlo para calcular el porcentaje de defectuosos p .

En cada iteración se generan \mathbf{M} observaciones aleatorias de una normal multivariante $N_k(\boldsymbol{\mu}_x^{(I)}, \Sigma_x^{(II)*})$, siendo \mathbf{M} un número suficientemente grande. Se seleccionan de esas observaciones aquéllas que estén dentro de la región rectangular de tolerancias T_x , que será obtenida como se describe a continuación.

Obtención de la región de tolerancias T_x .

Al igual que en el caso de independencia, donde los límites de tolerancias eran expresados como una distancia desde la media, en este caso la región de tolerancia T_x será un hiperrectángulo de dimensión k , centrado en la media y con caras paralelas a \mathbf{X} .

Para obtener esta región rectangular de tolerancias T_x , dada una matriz de covarianzas $\Sigma_X^{(II)}$ y un vector de medias $\mu_X^{(I)}$, se usará la misma metodología de proyección de un hiperelipsoide en un hiperrectángulo, descrita en el capítulo II.

Aplicando esta metodología las tolerancias t_i quedarán definidas como:

$$t_i = \sqrt{(\Sigma_X^{(II)})^{-1}_{ii} \chi_{k,1-\alpha}^2} \tag{4.13}$$

donde el subíndice ii denota el i -ésimo elemento de la diagonal de la inversa de la matriz $\Sigma_X^{(II)}$ y $\chi_{k,1-\alpha}^2$ el punto $100(1-\alpha)\%$ de la distribución Chi-cuadrado con k grados de libertad.

Los límites de tolerancia para cada variable X_i serán definidos como.

$$X_i \in [LTI_i, LTS_i] = [\mu_i^{(I)} - t_i, \mu_i^{(I)} + t_i] \tag{4.14}$$

La figura 4.2 muestra un ejemplo en dos dimensiones que representa una elipse definida por $\Sigma_X^{(II)}$ y $\mu_X^{(I)}$, y la región de tolerancia T_x obtenida mediante (4.13) y (4.14).

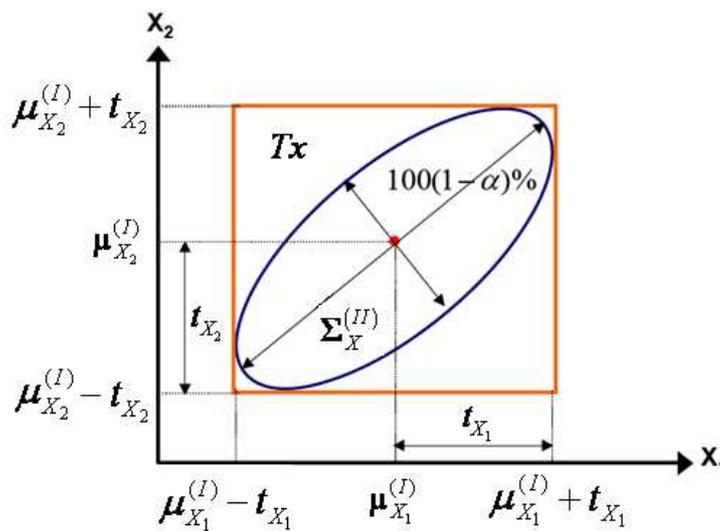


Figura 4.2: Región de tolerancia T_x para una matriz $\Sigma_X^{(II)}$ y un vector $\mu_X^{(I)}$.

Con las observaciones de \mathbf{X} seleccionadas, que denotaremos como \mathbf{X}_T se calculan las correspondientes variables \mathbf{Y}_T , aplicando la relación $\mathbf{Y}_T = \mathbf{A}\mathbf{X}_T$. Con estas observaciones \mathbf{Y}_T se calcula la proporción de defectuosos p :

$$p \equiv P(\mathbf{Y}_T \notin \mathcal{S} | \mathbf{Y}_T = f(\mathbf{X}_T)) \quad (4.15)$$

Si la proporción de defectuosos p es igual al valor α , entonces se detiene la búsqueda y la región de tolerancias de esa iteración será la solución del problema. En caso contrario, se sigue iterando.

Es importante mencionar aquí, que en esta segunda etapa, la proporción de defectuosos se calcula contando el número de observaciones que no cumplen las especificaciones de la pieza. Al igual que en la etapa I, en esta segunda etapa se podrá calcular la proporción de defectuosos considerando las especificaciones unilaterales, en los casos que sea así. De esta forma, se realizará un cálculo más preciso de la proporción de defectuosos y por consiguiente la asignación de tolerancias será más precisa.

También en esta etapa hay que tener en cuenta que se usará la relación real entre las variables \mathbf{Y} y \mathbf{X} . Es decir, en el caso en el que ésta relación fuese no lineal, y haya sido linealizada para obtener la matriz \mathbf{A} usada en la Etapa I, en esta etapa se usará la función no lineal entre las variables.

Finalizada la Etapa II obtendremos como resultado una nueva matriz de covarianzas $\Sigma_X^{(II)*}$, centrada en $\mu_x^{(I)}$, y el valor de las tolerancias óptimas de \mathbf{X} , tales que, LTI_i y LTS_i , objetivo de esta metodología.

Como se ve, en esta segunda etapa el problema de maximización de tolerancias se traduce en un problema de optimización univariante donde la variable de optimización es el coeficiente b . Este problema puede ser resuelto con algoritmos sencillos como por ejemplo el método de la secante y el método de bisección, que son los que se han utilizado en este proyecto. (Ver anexos [A.4] y [A.5]).

Es importante recordar, que en esta Etapa II, ya no se optimizan los valores nominales. Estos valores son los obtenidos en la Etapa I. La razón de ello es que al utilizar simulaciones de Monte Carlo, el tiempo de ejecución de cada iteración es alto y resolver un problema de optimización multivariante (más de 1 variable a optimizar) con AG no resulta factible.

A continuación se resumen brevemente los pasos a seguir en esta Etapa II:

1. Inicializamos la iteración mediante $r = 0$.
2. Partimos de la matriz de covarianzas $\Sigma_x^{(I)}$, y del vector de medias $\mu_x^{(I)}$, obtenidos en la última iteración de la Etapa I.
3. Aplicamos Monte Carlo, generando M observaciones aleatorias de una normal multivariante $N_k(\mu_x^{(I)}, \Sigma_x^{(I)})$ y seleccionando los valores \mathbf{X}_T que se encuentren dentro de la región de tolerancias T_x , definida por (4.13) y (4.14).
4. Dada la relación establecida entre \mathbf{X} e \mathbf{Y} , obtenemos los valores de \mathbf{Y}_T , mediante $\mathbf{Y}_T = \mathbf{A}\mathbf{X}_T$ y calculamos la proporción de defectuosos p establecida en (4.15).
5. Si $p = \alpha$, se detiene la búsqueda y la región de tolerancias obtenida como $[LTI_i, LTS_i] = [\mu_i^{(I)} - t_i, \mu_i^{(I)} + t_i]$, será la solución del problema. En caso contrario pasamos al paso 6.
6. Ponemos $r = r + 1$, definimos un coeficiente b y modificamos la matriz de covarianzas según el procedimiento visto en la sección, para encontrar unos nuevos valores, tales que, $\Sigma_x^{(II)*}$ y volvemos al paso 3 para continuar con la iteración.

5. CAPÍTULO QUINTO: EJEMPLOS DE APLICACIÓN DE LA METODOLOGÍA

En este capítulo se presentan tres ejemplos que permiten entender mejor la metodología propuesta en el capítulo IV. Para resolver el problema de optimización se usarán los algoritmos genéticos vistos en el capítulo III.

La utilización de algoritmos genéticos en el diseño de tolerancias y valores nominales se debe a que es un método global y robusto de búsqueda de soluciones cuya ventaja principal es el equilibrio entre eficiencia y eficacia para resolver problemas complejos de grandes dimensiones.

La resolución del problema de optimización aplicando la metodología propuesta en el capítulo IV puede ser compleja debido al alto número de variables que puede involucrar. En cada etapa de la metodología será necesario resolver un problema de optimización. En la Etapa I, por cada variable X_i tendremos 2 variables a optimizar, los factores b_i y g_i . En la Etapa II, se tendrá que optimizar un único factor b , que será el mismo para todas las variables X_i del problema.

Los problemas serán resueltos también considerando únicamente la asignación de tolerancias pero manteniendo constantes los valores nominales establecidos en la etapa de diseño. Esto nos permitirá comparar los resultados de ambas estrategias.

5.1. ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES EN UNA PIEZA: DEPÓSITO CILÍNDRICO.

El problema que se va a desarrollar fue propuesto en Yang y Naikan (2003). Estos autores resolvieron el problema de asignación de tolerancias teniendo en cuenta los requerimientos funcionales y los costes de fabricación asociados a los procesos productivos empleados, y considerando independencia entre variables.

En nuestro caso se usará este ejemplo para asignar tolerancias y valores nominales suponiendo correlación entre variables. Por este motivo, se modificarán y/o incluirán algunos datos, con el objetivo de adecuarlos al problema que se pretende resolver.

En la figura 5.1, se muestra el depósito cilíndrico compuesto por dos volúmenes internos.

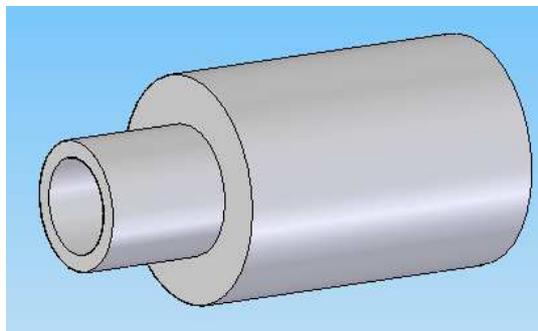


Figura 5.1: Depósito cilíndrico.

En este primer ejemplo de aplicación se desarrollarán los pasos más importantes para la aplicación de la metodología sobre una pieza independiente, correspondiente a un depósito cilíndrico, se mostrará la linealización de las variables mediante el desarrollo de Taylor, la aplicación de algoritmos genéticos como herramienta de optimización y se incluirá la programación, realizada en Matlab, utilizada para la resolución del problema, dando una definición breve de las funciones empleadas.

En la figura 5.2, se representan las variables $\mathbf{X} = R_1, R_2, R_3, R_4, L_1, L_2, L_3$, que definen el problema.

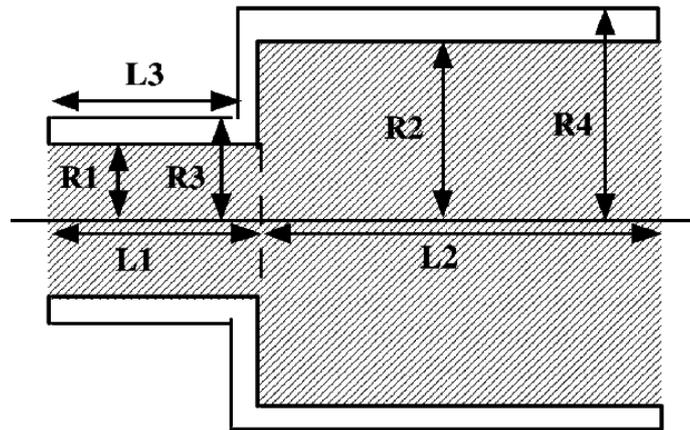


Figura 5.2: Variables \mathbf{X} .

Para la aplicación de la metodología se considera que en la etapa de diseño de la pieza se han definido valores nominales de las variables \mathbf{X} , que denotaremos como VN_d .

$$VN_d = 140, 190, 150, 200, 100, 200, 95$$

Las variables $\mathbf{Y} = V, T_1, T_2, T_3$ de la figura 5.3, definen las especificaciones funcionales de la pieza.

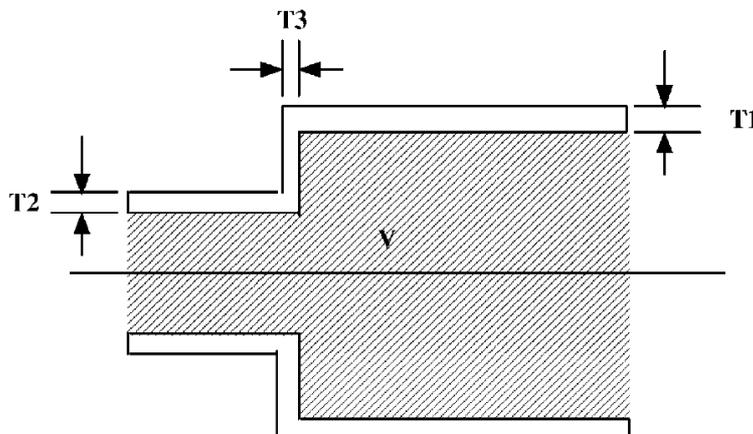


Figura 5.3: Variables \mathbf{Y} .

La variable V define el volumen interior total del depósito y las variables T_1, T_2, T_3 definen el espesor de las paredes del mismo en cada una de sus secciones. Estas variables son combinación de las variables \mathbf{X} de la forma que sigue a continuación.

$$\begin{aligned} V &= \pi R_1^2 L_1 + \pi R_2^2 L_2 \\ T_1 &= R_4 - R_2 \\ T_2 &= R_3 - R_1 \\ T_3 &= L_1 - L_3 \end{aligned}$$

Como se puede ver, la relación entre la variable V con las variables \mathbf{X} no es lineal por lo que será necesario aplicar un desarrollo de Taylor de primer orden que nos permita linealizar la relación (ver Capítulo II). Para ello usaremos los valores nominales establecidos en la etapa de diseño de la pieza, es decir:

$$\bar{R}_1 = 140 \text{ mm}, \bar{L}_1 = 100 \text{ mm}, \bar{R}_2 = 190 \text{ mm}, \bar{L}_2 = 200 \text{ mm}$$

Por tanto el punto alrededor del cual realizaremos el desarrollo de Taylor es:

$$\bar{\mathbf{X}} = 140, 190, 100, 200$$

Aplicando el desarrollo de Taylor de primer orden a la relación

$$V = \pi R_1^2 L_1 + \pi R_2^2 L_2$$

Tenemos que:

$$V = V \bar{\mathbf{X}} + \left. \frac{\partial V}{\partial R_1} \right|_{\bar{\mathbf{X}}} (R_1 - 140) + \left. \frac{\partial V}{\partial L_1} \right|_{\bar{\mathbf{X}}} (L_1 - 100) + \left. \frac{\partial V}{\partial R_2} \right|_{\bar{\mathbf{X}}} (R_2 - 190) + \left. \frac{\partial V}{\partial L_2} \right|_{\bar{\mathbf{X}}} (L_2 - 200)$$

Resolviendo:

$$V \bar{\mathbf{X}} = \pi \cdot 140^2 \cdot 100 + \pi \cdot 190^2 \cdot 200 = 2,8839 \cdot 10^6 \text{ mm}^3$$

$$\left. \frac{\partial V}{\partial R_1} \right|_{\bar{\mathbf{X}}} = 2R_1\pi L_1 \Big|_{\bar{\mathbf{X}}} = 8,79645 \cdot 10^3$$

$$\left. \frac{\partial V}{\partial L_1} \right|_{\bar{\mathbf{X}}} = R_1^2\pi \Big|_{\bar{\mathbf{X}}} = 6,1575 \cdot 10^3$$

$$\left. \frac{\partial V}{\partial R_2} \right|_{\bar{\mathbf{X}}} = 2R_2\pi L_2 \Big|_{\bar{\mathbf{X}}} = 2,38761 \cdot 10^4$$

$$\left. \frac{\partial V}{\partial L_2} \right|_{\bar{\mathbf{X}}} = R_2^2\pi \Big|_{\bar{\mathbf{X}}} = 1,13411 \cdot 10^4$$

Tras realizar las derivadas parciales obtenemos la siguiente relación lineal:

$$V = 8,79645 \cdot 10^3 R_1 + 6,1575 \cdot 10^3 L_1 + 2,38761 \cdot 10^4 R_2 + 1,13411 \cdot 10^4 L_2 - 5,768032 \cdot 10^6$$

Luego, las variables \mathbf{Y} son finalmente:

$$V = 8,79 \cdot 10^3 R_1 + 6,16 \cdot 10^3 L_1 + 2,39 \cdot 10^4 R_2 + 1,13 \cdot 10^4 L_2 - 5,77 \cdot 10^6$$

$$T_1 = R_4 - R_2$$

$$T_2 = R_3 - R_1$$

$$T_3 = L_1 - L_3$$

Por tanto, el vector $\mathbf{a} = a_{01}, a_{02}, \dots, a_{0j}$ ', y la matriz de relación \mathbf{A} son las siguientes:

$$\mathbf{a} = -5,77 \cdot 10^6, 0, 0, 0$$

$$\mathbf{A} = \begin{bmatrix} 8,79 \cdot 10^3 & 2,39 \cdot 10^4 & 0 & 0 & 6,16\pi \cdot 10^3 & 1,13 \cdot 10^4 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

Los límites de especificación para las variables \mathbf{Y} son:

$$2,8 \times 10^7 \text{ mm}^3 < V < 3,0 \times 10^7 \text{ mm}^3$$

$$9 \text{ mm} < T_1 < 11 \text{ mm}$$

$$9 \text{ mm} < T_2 < 11 \text{ mm}$$

$$4.5 \text{ mm} < T_3 < 5.5 \text{ mm}$$

Función objetivo y restricciones del problema

1. La función objetivo es maximizar la suma de las tolerancias t_i , $i = 1, 2, \dots, 7$ de las variables \mathbf{X} . Como en AG resolveremos el problema de minimización equivalente, la función objetivo queda definida como:

$$FO: \min z = -\sum_{i=1}^7 t_i$$

2. Se establecen unos valores mínimos para las tolerancias, $t_i \geq T_{i_min}$, de las variables \mathbf{X} , tal que: $T_{i_min} \geq 0.01$.

3. Los valores nominales de las variables \mathbf{X} , estarán comprendidos en un intervalo VNI_i, VNS_i , $i = 1, 2, \dots, 7$, donde:

$$VNI_i = (1 - f) * VN_{d(i)}$$

$$VNS_i = (1 + f) * VN_{d(i)}$$

Siendo $f = 5\%$

4. La proporción de defectuosos deseada será: $p \leq \alpha$, siendo $\alpha = 0,0027$

Programación en Matlab 7.0. Ver anexo [B.1].

Programa de Matlab para resolver la Etapa I: “*DepositoEtapa1.m*”.

Para resolver el problema partiremos de una matriz de covarianzas Σ_x generada al azar. En la práctica estos valores deben ser obtenidos del proceso de fabricación trabajando bajo control.

Esta matriz será el punto de partida de la metodología y se guardará dentro del fichero “*CovIniciales.mat*” para su ejecución en Matlab.

La matriz generada para este ejemplo quedaría como:

```
covarianzasx =
0.0009    0.0009    0.0002    0.0005    0.0004    0.0008    0.0007
0.0009    0.0031    0.0002    0.0005    0.0004    0.0008    0.0006
0.0002    0.0002    0.0024    0.0001    0.0001    0.0002    0.0002
0.0005    0.0005    0.0001    0.0009    0.0003    0.0005    0.0004
0.0004    0.0004    0.0001    0.0003    0.0025    0.0004    0.0003
0.0008    0.0008    0.0002    0.0005    0.0004    0.0013    0.0006
0.0007    0.0006    0.0002    0.0004    0.0003    0.0006    0.0028
```

En Matlab, esta matriz se define en las siguientes líneas del programa *DepositoEtapa1.m*:

```
load CovIniciales
covarianzasx=zeros(7,7);
covarianzasx=MatCov1;
a=length(covarianzasx);
```

Se considerará como valores nominales iniciales VN , los valores establecidos en la etapa de diseño VNd , de manera que a partir de ahora denotaremos:

$$VN = VN_d = 140, 190, 150, 200, 100, 200, 95$$

Como se vio en el capítulo IV, para realizar la asignación óptima de tolerancias y valores nominales se modificarán los valores de la matriz de covarianzas y el vector de valores nominales iniciales VN . Para ello, en esta primera etapa, se optimizarán mediante la aplicación de algoritmos genéticos los valores de los factores denominados “factor” y “factor2” correspondientes a los vectores $B = b_1, b_2 \cdots b_8$ y $G = g_1, g_2 \cdots g_8$, respectivamente.

Aplicación de Algoritmos Genéticos en la Etapa 1.

La aplicación de algoritmos genéticos se realizará mediante el programa de Matlab denominado “*DepositoEtapa1AG.m*”, donde se optimizará el valor de los factores b_i y g_i de cada variable X_i .

Dado que los valores que manejaremos serán decimales, la codificación de las variables del problema se realizará, como se vio en el capítulo III, mediante:

$$\frac{X_{\max} - X_{\min}}{2^L - 1} < m \Rightarrow L = \log_2 \left(1 + \frac{X_{\max} - X_{\min}}{m} \right)$$

Para los valores del vector B emplearemos:

$$\left. \begin{array}{l} x_{\max} = 8,1 \\ x_{\min} = 0,5 \\ m = 0,028 \end{array} \right\} L = \log_2 \left(1 + \frac{x_{\max} - x_{\min}}{m} \right) \approx 8$$

Para los valores del vector G emplearemos:

$$\left. \begin{array}{l} y_{\max} = 5 \\ x_{\min} = -5 \\ m = 0,038 \end{array} \right\} L = \log_2 \left(1 + \frac{y_{\max} - y_{\min}}{m} \right) \approx 8$$

Para ambos vectores mantendremos una longitud de cadena de ocho bits, es decir, el intervalo estará definido por $X_{\min} = 00000000$ y $X_{\max} = 11111111$. Por lo que podremos representar $2^8 = 256$ valores de cada variable comprendidos en el intervalo X_{\min}, X_{\max} .

Parámetros empleados para la resolución mediante algoritmos genéticos

1. Fitness función = @DepositoEtapa1AG. Que es el nombre de la función.
2. Number of variables = 112. Correspondientes a 2 vectores de 7 variables con 8bits.
3. Plots = Best fitness. Para que se vea cómo evoluciona la función objetivo.
4. Population type = Bit string. Se va a trabajar con números binarios.
5. Population size = 100. Utilizamos una población de entre 50 y 100 individuos.
6. Selection= Roulette.
7. Reproduction: crossover fraction = 0.6.
8. Mutación función: uniform. Rate = 0.9 a 0.03. Valor de probabilidad de mutación de cada bit.
Este ratio se irá modificando durante la ejecución del algoritmo según su evolución. En principio se designará un valor alto de 0,9 y se disminuirá según nos acerquemos a un valor pequeño hasta llegar a 0.03.
9. Crossover: Two point.
10. Stopping criteria: Generations: 200. Stall generation: 100. Stall time limit: 200.
11. Run solver.

Tras resolver el problema, obtendremos el mejor valor de la función fitness o función objetivo, que podemos ver representado en la figura 5.4, y la cadena de 112 bits correspondiente a los valores óptimos de los vectores B y G en código binario.

$$z = -3,43$$

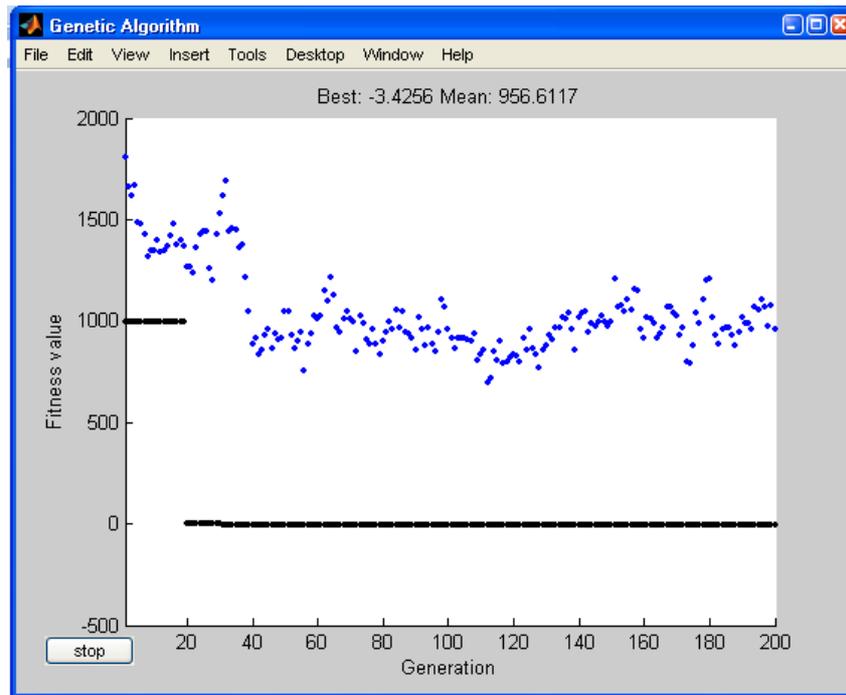


Figura 5.4: Plot Best fitness.

En la figura 5.4 se puede apreciar como en este caso, el algoritmo comienza la optimización penalizando a la función, estas penalizaciones serán impuestas para el cumplimiento de las restricciones del problema. Seguidamente, tras varias generaciones el algoritmo obtiene un valor que cumple con las restricciones establecidas, eliminando la penalización. La optimización continúa basándose en este valor y descartando las soluciones anteriores, continuando la optimización de forma correcta.

Los resultados obtenidos tras la primera etapa son la nueva matriz de covarianzas y el nuevo vector de medias de \mathbf{X} (o nuevo vector de valores nominales) necesarios para pasar a la Etapa2 de la metodología.

La nueva matriz de covarianzas es igual a:

`newcovarianzasx =`

0.0130	0.0124	0.0032	0.0079	0.0063	0.0116	0.0098
0.0124	0.0365	0.0041	0.0075	0.0112	0.0111	0.0164
0.0032	0.0041	0.0152	0.0019	0.0077	0.0029	0.0031
0.0079	0.0075	0.0019	0.0130	0.0038	0.0070	0.0059
0.0063	0.0112	0.0077	0.0038	0.0166	0.0057	0.0092
0.0116	0.0111	0.0029	0.0070	0.0057	0.0187	0.0088
0.0098	0.0164	0.0031	0.0059	0.0092	0.0088	0.0285

El nuevo vector de valores nominales o medias es igual a:

```
newmediax =
    137.8429
    188.1181
    148.2832
    197.7019
    100.9127
    203.8030
    95.9067
```

Una vez obtenidos estos valores pasamos a la segunda etapa en la que, como se vio en el capítulo IV, se ajustará la proporción de defectuosos mediante una simulación del método de Monte Carlo, visto en el capítulo IV.

Programa de Matlab para resolver la Etapa II: “DepositoEtapa2.m”.

En esta segunda etapa se aplicará el método de Monte Carlo, para ello partiendo de los resultados obtenidos en la primera etapa, se generarán 10^5 distribuciones normales multivariantes que serán comparadas con la región de tolerancias establecida por T_x . De las distribuciones que cumplan con los límites establecidos por dicha región se comprobarán las que cumplen con las especificaciones establecidas por las variables Y , obteniendo como objetivo la proporción de defectuosos final.

Es este ejemplo de aplicación y tras realizar la Etapa II, los resultados finales que se han obtenido son:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de las tolerancias de X:

$$\sum_{i=1}^7 t_i = 4,53$$

Tolerancias y valores nominales de X:

$$R_1 = 137.85 \pm 0.53 \text{ mm}$$

$$R_2 = 188.12 \pm 0.89 \text{ mm}$$

$$R_3 = 148.28 \pm 0.58 \text{ mm}$$

$$R_4 = 197.7 \pm 0.53 \text{ mm}$$

$$L_1 = 100.9 \pm 0.60 \text{ mm}$$

$$L_2 = 203.8 \pm 0.64 \text{ mm}$$

$$L_3 = 95.9 \pm 0.79 \text{ mm}$$

Comparación de asignación de tolerancias y valores nominales vs asignación de tolerancias

Se compararán los resultados obtenidos arriba con los que se obtienen a partir de la asignación de tolerancias manteniendo constante el valor de los valores nominales de las variables X . Estos valores nominales serán los establecidos en la etapa de diseño:

$$VN = VN_d = 140, 190, 150, 200, 100, 200, 95$$

Los resultados obtenidos son en este caso:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de tolerancias de X:

$$\sum_{i=1}^7 t_i = 3.75$$

Tolerancias de X:

$$R_1 = 140 \pm 0.39 \text{ mm}$$

$$R_2 = 190 \pm 0.7 \text{ mm}$$

$$R_3 = 150 \pm 0.58 \text{ mm}$$

$$R_4 = 200 \pm 0.39 \text{ mm}$$

$$L_1 = 100 \pm 0.57 \text{ mm}$$

$$L_2 = 200 \pm 0.46 \text{ mm}$$

$$L_3 = 95 \pm 0.66 \text{ mm}$$

Conclusión.

En este primer ejemplo podemos ver como la aplicación de la metodología propuesta en este proyecto, donde se considera la asignación de tolerancias y valores nominales, obtiene una suma de tolerancias mayor, es decir, la maximización de la función objetivo es mejor que la obtenida únicamente con la asignación de tolerancias donde los valores nominales son considerados fijos.

También podemos ver como mejora la tolerancia individual de cada variable y como el valor medio de éstas varía dentro de los límites establecidos con anterioridad de manera aceptable.

5.2. ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES EN UN ENSAMBLE DE BLOQUES.

El problema que se va a desarrollar fue propuesto en los trabajos de Lee y Wood (1990), y Lee y Johnson (1993). Estos autores resolvieron el problema de asignación de tolerancias bajo la suposición de independencia. En González y Sánchez (2008), este ejemplo fue utilizado para resolver el problema de asignación de tolerancias bajo la suposición de correlación entre variables. Aquí se usará este ejemplo para asignar tolerancias y valores nominales suponiendo correlación entre variables. Por este motivo, se modificarán e incluirán algunos de los datos dados en los artículos, con el objetivo de adecuarlos al problema que se pretende resolver.

La figura 5.5, muestra el ensamblaje compuesto por 2 piezas independientes.

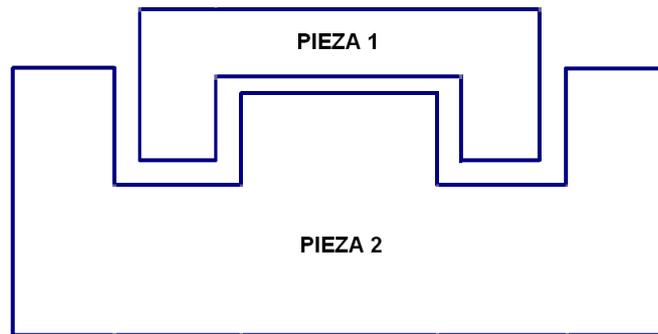


Figura 5.5: Ensamblaje de Bloques.

En este ejemplo de aplicación se desarrollarán los pasos más importantes para la aplicación de la metodología en un ensamblaje mecánico, se mostrará la aplicación de algoritmos genéticos como herramienta de optimización y se incluirá la programación, realizada en Matlab, utilizada para la resolución del problema, dando una definición breve de las funciones empleadas.

En la figura 5.6, se representan las variables $\mathbf{X} = X_1, X_2, \dots, X_8$, que definen el problema.

En este caso, las dos piezas independientes suman un total de ocho variables.

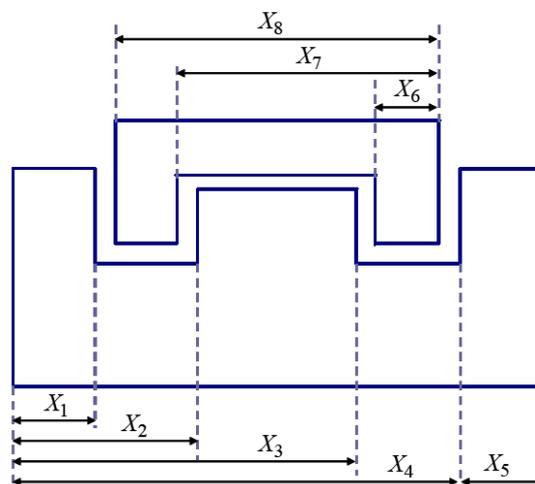


Figura 5.6: Asignación de variables. Bloques.

Para la aplicación de la metodología se considera que en la etapa de diseño de la pieza se han definido los valores nominales de las variables \mathbf{X} , que denotaremos como VNd .

$$VNd = 25.4, 50.8, 76.2, 101.6, 25.4, 25.3, 50.8, 76.1$$

Las variables $\mathbf{Y} = Y_1, Y_2, \dots, Y_j$, son combinación lineal de las variables \mathbf{X} siendo:

$$Y_1 = X_4 + X_5.$$

$$Y_2 = X_2 - X_1 - X_8 + X_7.$$

$$Y_3 = X_7 - X_6 - X_3 + X_2.$$

$$Y_4 = X_4 - X_3 - X_6.$$

Estas variables \mathbf{Y} definen las características funcionales del ensamblaje mecánico, la variable Y_1 define la longitud total del conjunto y las variables Y_2, Y_3, Y_4 las dimensiones de holgura necesarias para un correcto ensamblaje.

Los límites de especificación para estas variables son:

$$Y_1 \leq 127.13 \text{ mm.}$$

$$Y_2 \geq 0.0076 \text{ mm.}$$

$$Y_3 \geq 0.0254 \text{ mm.}$$

$$Y_4 \geq 0.0076 \text{ mm.}$$

El vector \mathbf{a} es un vector nulo y la matriz de relación \mathbf{A} queda entonces definida como:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

La función objetivo y las restricciones del problema son iguales a las usadas en el problema anterior.

Programación en Matlab 7.0. Ver anexo [B.2].**Programa de Matlab para resolver la Etapa I: BloquesEtapa1.m.**

Para resolver el problema partiremos de una matriz de covarianzas Σ_X generada al azar. En la práctica estos valores deben ser obtenidos del proceso de fabricación trabajando bajo control. En el caso de este ejemplo, cada una de las piezas que forman el conjunto tiene su correspondiente matriz de covarianzas. Una vez generadas por separado, las unimos en una única matriz de covarianzas inicial Σ_X^0 dividida en dos partes, la primera parte contendrá los valores correspondientes a la pieza uno, cuyas variables son X_1, X_2, X_3, X_4, X_5 y la segunda parte recogerá los valores de las variables correspondientes a la segunda pieza X_6, X_7, X_8 , formando la matriz completa.

Esta matriz será el punto de partida de la metodología y se guardará dentro del fichero "CovIniciales.mat" para su ejecución en Matlab.

La matriz obtenida a partir de las 2 matrices aleatorias generadas quedaría como:

```
covarianzasx =
    0.0102    0.0045   -0.0000    0.0083    0.0035         0         0         0
    0.0045    0.0042   -0.0001    0.0037    0.0026         0         0         0
   -0.0000   -0.0001    0.0023    0.0001   -0.0002         0         0         0
    0.0083    0.0037    0.0001    0.0188    0.0029         0         0         0
    0.0035    0.0026   -0.0002    0.0029    0.0804         0         0         0
         0         0         0         0         0    0.0026    0.0016    0.0020
         0         0         0         0         0    0.0016    0.0011    0.0013
         0         0         0         0         0    0.0020    0.0013    0.0019
```

En Matlab, esta matriz se define en las siguientes líneas del programa BloquesEtapa1.m:

```
load CovIniciales

covarianzasx=zeros(8,8);
covarianzasx(1:5,1:5)=MatCov1;
covarianzasx(6:8,6:8)=MatCov2;
a=length(covarianzasx);
```

Se considerará como valores nominales iniciales VN , los valores establecidos en la etapa de diseño VNd , de manera que a partir de ahora denotaremos:

$$VN = VN_d = 25.4, 50.8, 76.2, 101.6, 25.4, 25.3, 50.8, 76.1$$

Como se vio en el capítulo IV, para realizar la asignación óptima de tolerancias y valores nominales se modificarán los valores de la matriz de covarianzas y el vector de valores nominales iniciales VN . Para ello, en esta primera etapa, se optimizarán mediante la aplicación de algoritmos genéticos los valores de los factores denominados "factor" y "factor2" correspondientes a los vectores $B = b_1, b_2 \cdots b_8$ y $G = g_1, g_2 \cdots g_8$, respectivamente.

Aplicación de Algoritmos Genéticos en la Etapa 1.

La aplicación de algoritmos genéticos se realizará mediante el programa de Matlab denominado “*BloquesEtapa1AG.m*”, donde se empleara la toolbox “*gatool*”.

Codificación de las variables:

Dado que los valores que manejaremos serán decimales, la codificación de las variables del problema se realizará, como se vio en el capítulo III, mediante:

$$\frac{X_{\max} - X_{\min}}{2^L - 1} < m \Rightarrow L = \log_2 \left(1 + \frac{X_{\max} - X_{\min}}{m} \right)$$

Para los valores del vector B emplearemos:

$$\left. \begin{array}{l} x_{\max} = 17 \\ x_{\min} = 0.3 \\ m = 0,07 \end{array} \right\} L = \log_2 \left(1 + \frac{x_{\max} - x_{\min}}{m} \right) \approx 8$$

Para los valores del vector G emplearemos:

$$\left. \begin{array}{l} y_{\max} = 2 \\ x_{\min} = -5 \\ m = 0,028 \end{array} \right\} L = \log_2 \left(1 + \frac{y_{\max} - y_{\min}}{m} \right) \approx 8$$

Para ambos vectores mantendremos una longitud de cadena de ocho bits, es decir, el intervalo estará definido por $X_{\min} = 00000000$ y $X_{\max} = 11111111$. Por lo que podremos representar $2^8 = 256$ valores de cada variable comprendidos en el intervalo X_{\min}, X_{\max} , obteniendo una precisión mayor a la especificada.

Parámetros empleados para la resolución mediante algoritmos genéticos

1. Fitness función = @BloquesEtapa1AG. Que es el nombre de la función.
2. Number of variables = 128. Correspondientes a 2 vectores de 8 variables con 8bits.
3. Plots = Best fitness. Para que se vea cómo evoluciona la función objetivo.
4. Population type = Bit string. Se va a trabajar con números binarios.
5. Population size = 100. Utilizamos una población de entre 50 y 100 individuos.
6. Selection= Roulette.
7. Reproduction: crossover fraction = 0.6.
8. Mutación función: uniform. Rate = 0.9 a 0.03. Valor de probabilidad de mutación de cada bit.
Este ratio se irá modificando durante la ejecución del algoritmo según su evolución. En principio se designará un valor alto de 0,9 y se disminuirá según nos acerquemos a un valor pequeño hasta llegar a 0.03.
9. Crossover: single point.
10. Stopping criteria: generations: 200, stall generation: 100, stall time limit: 200.
11. Run solver.

Tras resolver el problema, obtendremos el mejor valor de la función fitness o función objetivo, que podemos ver representado en la figura 5.7, y la cadena de 128 bits correspondiente a los valores óptimos de los vectores B y G en código binario.

$$z = -10,47$$

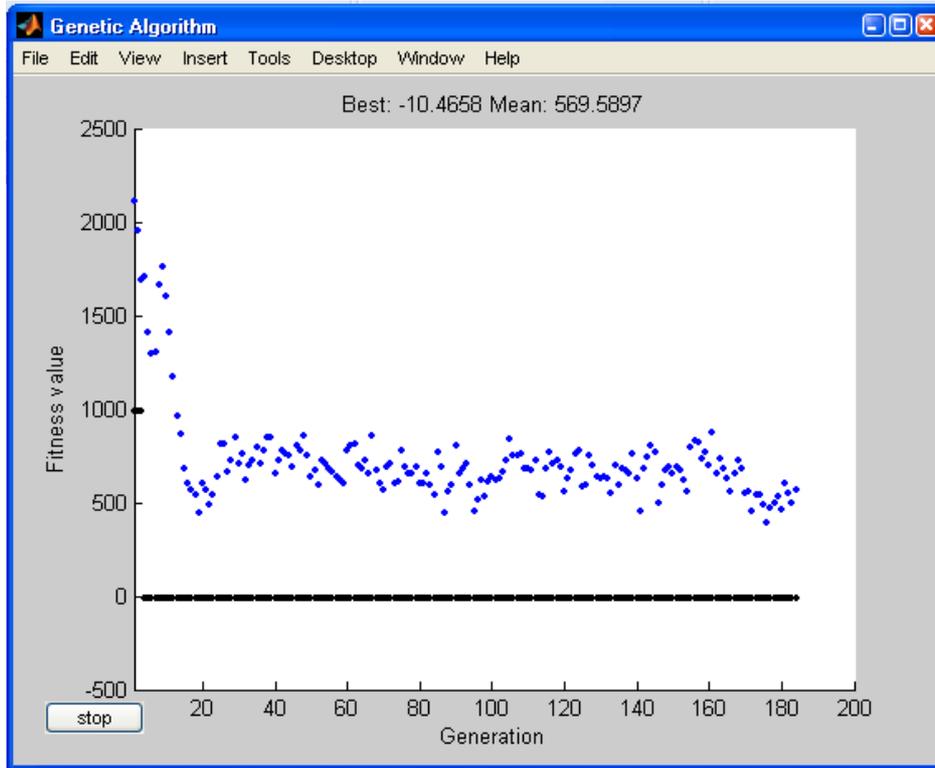


Figura 5.7: Plot Best fitness.

En la figura 5.7 se puede apreciar como en este caso, el algoritmo comienza con una penalización breve, es decir, encuentra de forma rápida una cadena de 128 bits que cumple con las restricciones de la función. En este caso, la optimización continúa optimizando la función objetivo basándose en este valor y descartando posibles soluciones externas a este camino.

Los resultados obtenidos tras la primera etapa son la nueva matriz de covarianzas y el nuevo vector de medias de \mathbf{X} (o nuevo vector de valores nominales) necesarios para pasar a la Etapa2 de la metodología.

La nueva matriz de covarianzas es igual a:

newcovarianzasx =

0.0347	0.0173	-0.0002	0.0239	0.0778	0	0	0
0.0173	0.0145	-0.0004	0.0110	0.0538	0	0	0
-0.0002	-0.0004	0.0083	0.0001	-0.0030	0	0	0
0.0239	0.0110	0.0001	0.0569	0.0743	0	0	0
0.0778	0.0538	-0.0030	0.0743	1.3645	0	0	0
0	0	0	0	0	0.0201	0.0127	0.0168
0	0	0	0	0	0.0127	0.0081	0.0105
0	0	0	0	0	0.0168	0.0105	0.0142

El vector de valores nominales o medias de X es igual a:

```
newmediax =
    24.2055
    48.9882
    72.5102
    97.8126
    24.2273
    24.0673
    48.9023
    73.0787
```

Una vez obtenidos estos valores pasamos a la segunda etapa en la que, como se vio en el capítulo IV, se ajustará la proporción de defectuosos mediante una simulación del método de Monte Carlo.

Programa de Matlab para resolver la Etapa II : “*BloquesEtapa2.m*”.

En esta segunda etapa se aplicará el método de Monte Carlo, para ello partiendo de los resultados obtenidos en la primera etapa, se generarán 10^5 distribuciones normales multivariantes que serán comparadas con la región de tolerancias establecida por T_x . De las distribuciones que cumplan con los límites establecidos por dicha región se comprobarán las que cumplen con las especificaciones establecidas por las variables Y , obteniendo como objetivo la proporción de defectuosos final.

Es este ejemplo de aplicación y tras realizar la Etapa II, los resultados finales que se han obtenido son:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de las tolerancias de X:

$$\sum_{i=1}^8 t_i = 14,26$$

Tolerancias y valores nominales de X:

$$X_1 = 24.20 \pm 1.23 \text{ mm}$$

$$X_2 = 49 \pm 0.80 \text{ mm}$$

$$X_3 = 72.51 \pm 0.60 \text{ mm}$$

$$X_4 = 97.81 \pm 1.58 \text{ mm}$$

$$X_5 = 24.23 \pm 7.73 \text{ mm}$$

$$X_6 = 24.10 \pm 0.94 \text{ mm}$$

$$X_7 = 48.90 \pm 0.60 \text{ mm}$$

$$X_8 = 73.10 \pm 0.79 \text{ mm}$$

Comparación de asignación de tolerancias y valores nominales vs asignación de tolerancias

Se compararán los resultados obtenidos arriba con los que se obtienen a partir de la asignación de tolerancias manteniendo constante el valor de los valores nominales de las variables \mathbf{X} . Estos valores nominales son los establecidos en la etapa de diseño:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de las tolerancias de \mathbf{X} :

$$\sum_{i=1}^8 t_i = 0,74$$

Tolerancias de \mathbf{X} :

$$X_1 = 25.4 \pm 0.1245 \text{ mm}$$

$$X_2 = 50.8 \pm 0.1011 \text{ mm}$$

$$X_3 = 76.2 \pm 0.0312 \text{ mm}$$

$$X_4 = 101.6 \pm 0.1111 \text{ mm}$$

$$X_5 = 25.4 \pm 0.1455 \text{ mm}$$

$$X_6 = 25.3 \pm 0.0809 \text{ mm}$$

$$X_7 = 50.8 \pm 0.0623 \text{ mm}$$

$$X_8 = 76.1 \pm 0.0803 \text{ mm}$$

Conclusión.

En este segundo ejemplo, al igual que en el primero, podemos ver como la aplicación de la metodología propuesta mejora los valores de maximización de tolerancias en comparación con los obtenidos únicamente con la asignación de tolerancias donde los valores nominales considerados fijos.

También podemos ver como mejora la tolerancia individual de cada variable y como el valor medio de éstas varía dentro de los límites establecidos con anterioridad de manera aceptable.

Es importante resaltar en este ejemplo la diferencia de valores obtenidos en la suma de tolerancias de las dos metodologías. Esta gran variación es debida a la falta de límites bilaterales, es decir, en este ejemplo, a diferencia del ejemplo anterior, sólo se han considerado restricciones unilaterales a las variables Y , lo cual provoca que se puedan dar altos valores de tolerancia en variables de baja repercusión en el ensamblaje como es el caso de la variable X_5 , la cual sólo toma parte en la variable que restringe la longitud total del ensamblaje.

5.3 ASIGNACIÓN DE TOLERANCIAS Y VALORES NOMINALES EN UN ENSAMBLE DE CILINDROS.

El problema de la figura 5.8, fue propuesto en el modelo de Weil basado en los métodos productivos y costes (Wei,1988; Zha,1992). En sus artículos estos autores asignan las tolerancias del ensamblaje en función de los métodos productivos y los costes. Aquí se usará este ejemplo para asignar tolerancias y valores nominales suponiendo correlación entre variables. Por este motivo, se modificarán e incluirán algunos de los datos dados en los artículos, con el objetivo de adecuarlos al problema que se pretende resolver.

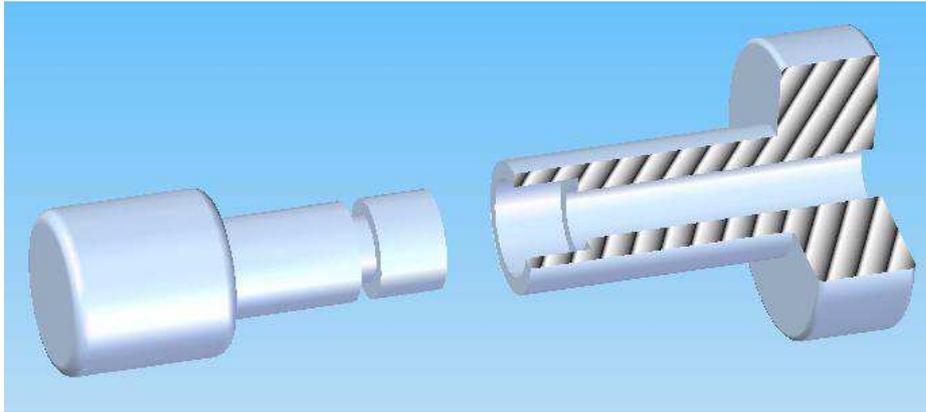


Figura 5.8: Ensamblaje mecánico. Cilindros.

Una vez desarrollados, en el ejemplo anterior, los pasos más importantes para la aplicación de la metodología, en este ejemplo se hará un desarrollo más reducido de la misma.

En la figura 5.9, se representan las variables $\mathbf{X} = X_1, X_2, \dots, X_7$, que definen el problema. En este caso, las dos piezas independientes suman un total de siete variables.

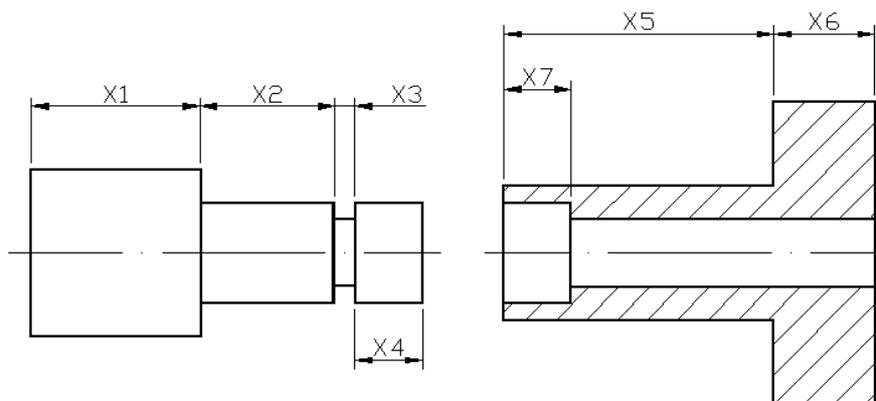


Figura 5.9: Asignación de variables. Cilindros.

Para la aplicación de la metodología se considera que en la etapa de diseño de la pieza se han definido los valores nominales de las variables \mathbf{X} , que denotaremos como VNd .

$$VN_d = 25, 20, 3, 10, 40, 15, 10$$

Las variables $\mathbf{Y} = Y_1, Y_2, \dots, Y_j$, son combinación lineal de las variables \mathbf{X} siendo:

$$Y_1 = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 - X_7$$

$$Y_2 = X_2 + X_3 + X_4 - X_7.$$

Estas variables \mathbf{Y} definen las características funcionales del ensamblaje mecánico, la variable Y_1 define la longitud total del conjunto y la variable Y_2 la longitud central del ensamble.

Los límites de especificación para estas variables son:

$$Y_1 \leq 103.5 \text{ mm.}$$

$$Y_2 \geq 22.8 \text{ mm.}$$

El vector \mathbf{a} es un vector nulo y la matriz de relación \mathbf{A} queda entonces definida como:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}$$

La función objetivo y las restricciones del problema son iguales a las usadas en el problema anterior.

Programación en Matlab 7.0. Ver anexo [B.3].

Programa de Matlab para resolver la Etapa I: “CilindrosEtapa1.m”.

Para resolver el problema partiremos de una matriz de covarianzas Σ_x generada al azar. En la práctica estos valores deben ser obtenidos del proceso de fabricación trabajando bajo control.

Esta matriz será el punto de partida de la metodología y se guardará dentro del fichero “CovIniciales.mat” para su ejecución en Matlab.

La matriz generada para este ejemplo quedaría como:

```

covarianzasx =
    0.0109    0.0023    0.0102    0.0052    0    0    0
    0.0023    0.0027    0.0022    0.0009    0    0    0
    0.0102    0.0022    0.0120    0.0049    0    0    0
    0.0052    0.0009    0.0049    0.0153    0    0    0
    0    0    0    0    0.1373    0.0587    0.0136
    0    0    0    0    0.0587    0.0253    0.0058
    0    0    0    0    0.0136    0.0058    0.0015

```

Se considerará como valores nominales iniciales VN , los valores establecidos en la etapa de diseño VN_d , de manera que a partir de ahora denotaremos:

$$VN = VN_d = 25, 20, 3, 10, 40, 15, 10$$

Como se vio en el capítulo IV, para realizar la asignación óptima de tolerancias y valores nominales se modificarán los valores de la matriz de covarianzas y el vector de valores nominales iniciales VN . Para ello, en esta primera etapa, se optimizarán mediante la aplicación de algoritmos genéticos los valores de los factores denominados “factor” y “factor2” correspondientes a los vectores $B = b_1, b_2 \dots b_8$ y $G = g_1, g_2 \dots g_8$, respectivamente.

La aplicación de algoritmos genéticos y los parámetros empleados para su resolución serán implementados en el programa de Matlab denominado “*CilindrosEtapa1AG.m*”, donde se empleará la toolbox “*gatool*”, siguiendo las pautas dadas en el ejemplo anterior.

Tras resolver el problema, obtendremos el mejor valor de la función fitness o función objetivo, que podemos ver representado en la figura 5.10, y la cadena de 112 bits correspondiente a los valores óptimos de los vectores B y G en código binario.

$$z = -5.71$$

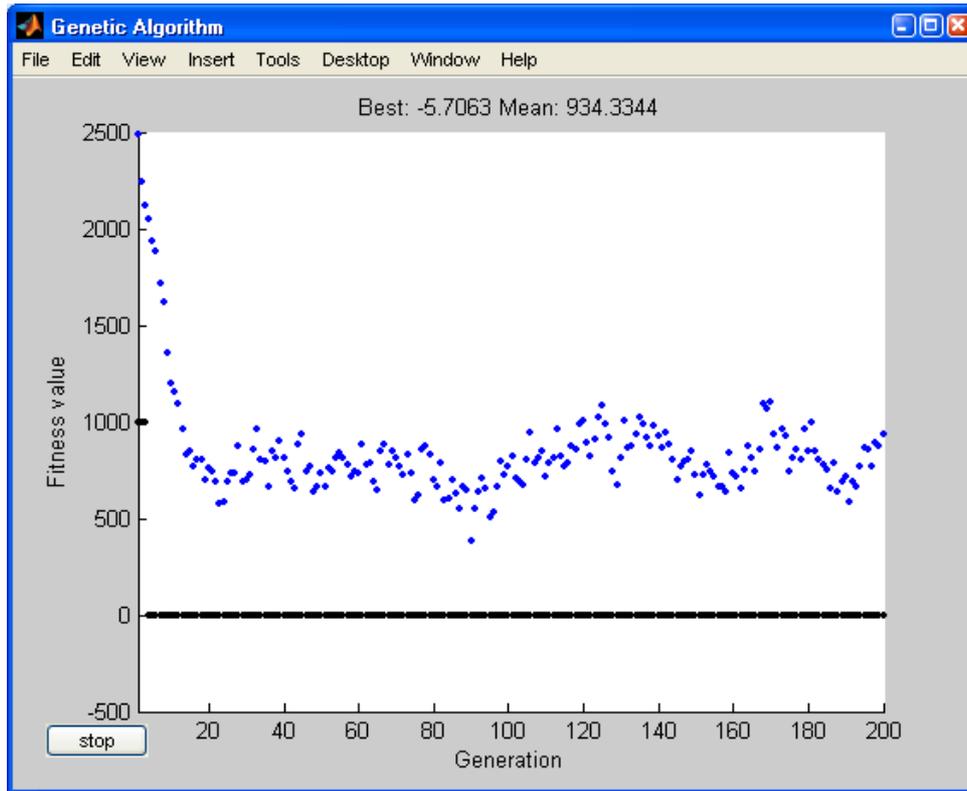


Figura 5.10: Plot Best fitness.

En la figura 5.10 se puede apreciar como en este caso, el algoritmo comienza la optimización penalizando, es decir, no encuentra de forma directa una cadena de 128 bits que cumpla con las restricciones de la función, pero pasadas una serie de iteraciones, el algoritmo deja de penalizar la función y continua la optimización de la función por un camino correcto.

La nueva matriz de covarianzas de \mathbf{X} obtenida en esta Etapa 1 es igual a:

`newcovarianzasx =`

0.0162	0.0028	0.0106	-0.0136	0	0	0
0.0028	0.0137	0.0019	-0.0049	0	0	0
0.0106	0.0019	0.0197	-0.0170	0	0	0
-0.0136	-0.0049	-0.0170	0.0508	0	0	0
0	0	0	0	0.1534	0.0652	0.0152
0	0	0	0	0.0652	0.0291	0.0064
0	0	0	0	0.0152	0.0064	0.0023

El nuevo vector de valores nominales o medias es igual a:

```
newmediax =
    24.2978
    20.2852
     3.0749
    10.1995
    39.3666
    14.2644
     9.9793
```

Una vez obtenidos estos valores pasamos a la segunda etapa en la que, como se vio en el capítulo IV, se ajustará la proporción de defectuosos mediante una simulación del método de Monte Carlo.

Programa de Matlab para resolver la Etapa II: "*CilindrosEtapa2.m*".

En esta segunda etapa se aplicará el método de Monte Carlo, para ello, partiendo de los resultados obtenidos en la primera etapa, se generarán 10^5 distribuciones normales multivariantes que serán comparadas con la región de tolerancias establecida por T_x . De las distribuciones que cumplan con los límites establecidos por dicha región se comprobarán las que cumplen con las especificaciones establecidas por las variables Y , obteniendo como objetivo la proporción de defectuosos final.

Es este ejemplo de aplicación y tras realizar la Etapa II, los resultados finales que se han obtenido son:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de las tolerancias de X:

$$\sum_{i=1}^7 t_i = 6,7$$

Tolerancias y valores nominales de X:

$$X_1 = 24.30 \pm 0.7 \text{ mm}$$

$$X_2 = 20.28 \pm 0.64 \text{ mm}$$

$$X_3 = 3.07 \pm 0.77 \text{ mm}$$

$$X_4 = 10.2 \pm 1.24 \text{ mm}$$

$$X_5 = 39.37 \pm 2.15 \text{ mm}$$

$$X_6 = 14.26 \pm 0.94 \text{ mm}$$

$$X_7 = 9.98 \pm 0.26 \text{ mm}$$

Comparación de asignación de tolerancias y valores nominales vs asignación de tolerancias

Se compararán los resultados obtenidos arriba con los que se obtienen a partir de la asignación de tolerancias manteniendo constante el valor de los valores nominales de las variables X . Estos valores nominales son los establecidos en la etapa de diseño:

Proporción de defectuosos:

$$p = 0,0027$$

Suma de las tolerancias de X:

$$\sum_{i=1}^7 t_i = 2.24$$

Tolerancias de X:

Asignación de tolerancias:

$$X_1 = 25 \pm 0.26 \text{ mm}$$

$$X_2 = 20 \pm 0.1 \text{ mm}$$

$$X_3 = 3 \pm 0.31 \text{ mm}$$

$$X_4 = 10 \pm 0.64 \text{ mm}$$

$$X_5 = 40 \pm 0.56 \text{ mm}$$

$$X_6 = 15 \pm 0.22 \text{ mm}$$

$$X_7 = 10 \pm 0.1 \text{ mm}$$

Conclusión.

Como en los anteriores ejercicios, en este tercer ejemplo podemos ver como la aplicación de la metodología propuesta mejora los valores de maximización de tolerancias en comparación con los obtenidos únicamente con la asignación de tolerancias donde los valores nominales considerados fijos.

También podemos ver como mejora la tolerancia individual de cada variable y como el valor medio de éstas varía dentro de los límites establecidos con anterioridad de manera aceptable.

Al igual que en el ejercicio anterior, en este ejemplo únicamente hemos restringido las variables **Y** unilateralmente pero en este caso, debido a la mayor dependencia entre variables, los valores de las tolerancias de **X** no son tan altos como ocurría en dicho ejercicio.

6. CAPÍTULO SEXTO: CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

6.1. CONCLUSIONES.

Se ha propuesto una metodología para la asignación óptima de tolerancias y valores nominales a un conjunto de variables correladas. Cumpliendo con uno de los objetivos del proyecto, se ha elaborado un documento que describe detalladamente esta metodología y sirve como guía para su aplicación. Este documento cubre los siguientes aspectos:

- Se describen detalladamente los conceptos importantes relacionados con la metodología que se propone para la asignación de tolerancias y valores nominales a una pieza o ensamblaje mecánico. Estos conceptos se han descrito de forma clara y sencilla en el capítulo II de este documento. Asimismo, se han descrito los métodos tradicionales para la asignación óptima de tolerancias, resaltando los aspectos en los que se basa la metodología que se propone aquí.
- Se describe de forma detallada, paso a paso, la metodología propuesta. Este desarrollo se ha realizado de manera que la aplicación de esta metodología no esté condicionada a tener altos conocimientos de estadística, es decir, se ha planteado de forma que pueda ser interpretada por cualquier usuario interesado en el estudio de nuevas metodologías para la asignación de tolerancias y valores nominales.
- Se incluye, en el capítulo III, una introducción a los algoritmos genéticos y su aplicación como herramienta de optimización. También se incluye una guía de utilización de la interfaz gráfica de la toolbox de algoritmos genéticos en MATLAB "gatool". Estos capítulos permiten que el uso de esta herramienta resulte sencillo para el usuario.
- Se incluyen diferentes ejemplos de aplicación de la metodología. En estos ejemplos se describe paso a paso la aplicación de la metodología incluyendo una clara explicación del planteamiento y formulación y una serie de figuras de gran ayuda para la comprensión del ejercicio. La resolución de los ejercicios se hace utilizando la toolbox de MATLAB para algoritmos genéticos y las conclusiones se obtienen mediante la comparativa de asignación de tolerancias y valores nominales vs asignación de tolerancias.

A partir de los resultados obtenidos en los ejemplos presentados, se puede concluir que la inclusión de la posibilidad de variar los valores nominales de las variables mejora la asignación óptima de tolerancias. También, se puede concluir que los Algoritmos Genéticos son una herramienta eficaz como método de optimización y su implementación en la toolbox de MATLAB una forma potente y rápida de resolución. Se ha podido comprobar que a pesar de que la complejidad del problema de optimización es mayor que la que se tiene considerando los valores nominales fijos, es factible obtener una solución mediante la aplicación de algoritmos genéticos.

6.2. FUTURAS LÍNEAS DE TRABAJO.

En este apartado se pretenden abrir nuevas líneas de investigación que ayuden a conseguir una mejora de la metodología mostrada en este proyecto.

1. Aplicar la metodología propuesta en este proyecto considerando distintos procesos de producción y diferentes costes relacionados con la asignación de tolerancias.
2. Asignar tolerancias y valores nominales considerando distribuciones de las variables distintas a la distribución normal.
3. Mejorar la aplicación de los algoritmos genéticos, mediante la utilización de otros tipos de codificación de las variables, como puede ser la codificación Gray. Este tipo de codificación permite eliminar posibles problemas relacionados con el código binario. En esta misma línea, aplicar otra herramienta de optimización, como por ejemplo aplicar Estrategias Evolutivas que no codifican las soluciones, si no que emplean valores reales.

BIBLIOGRAFÍA.

1. **Peña, D.** “Estadística. Modelos y métodos 1. Fundamentos”. Alianza. 2ªEdición. (1992).
2. **Peña, D.** “Estadística. Modelos y métodos. 2. Modelos lineales y series temporales”. Alianza. 2ªEdición. (2000).
3. **Peña, D.** “Análisis de datos multivariantes”. Mc Graw Hill. (2002).
4. **Cuadras, C. M.** “Nuevos métodos de análisis multivariante”. CMC Editions. (2007).
5. **Darwin, C.** “El origen de las especies”. Austral. 10ª Edición. (2009).
6. **González, I.** “Apuntes de la asignatura: Técnicas Avanzadas de Diseño en Ingeniería Mecánica”. Máster de Ingeniería de Máquinas y Transportes. (Curso 2009-2010).
7. **Paz, J.** “Proyecto fin de carrera: Aplicación de algoritmos genéticos a problemas de diseño en ingeniería mecánica utilizando Matlab”. (Curso: 2006-2007).
8. **Manual de Matlab.** “The Language of Technical Computing”, The MathWorks (2002).

ARTÍCULOS RELACIONADOS.

1. **Isabel González and Ismael Sánchez (2009)**. "Statistical tolerance synthesis with correlated variables". *Mechanism and Machine Theory* 44 (2009) 1097–1107. University Carlos III de Madrid.
2. **Montano, A.** (2008). "Algoritmos Genéticos en la discriminación". *Revista Investigación Operacional, Universidad de Veracruz*.
3. **Pierluissi, L. Claudio, M. Rocco, S.** (2007). "Optimal Design Centring Through a Hybrid Approach Base don Evolutionary Algorithms and Monte Carlo Simulation". *Universidad central de Venezuela*.
4. **Zielinski, L. Rutkowski, J.** (2006). " Design Centering and Tolerancing with Utilization of Evolutionary Techniques". *Universidad de Gliwice*.
5. **Noorul, A. Sivakumar, K.** (2005). "Tolerance design optimization of machine elements using genetic algorithm". *Advanced Manufacturing Technology*.
6. **Ramos, B. García, E. Peláez, J.** (2002). "Análisis de las Tolerancias de Ensamblaje Asistidas por Ordenador en el Diseño (CAT)". *Universidad de Burgos*.
7. **William, L. Wu, C.F.J.** (1999). "An Integrated Method of Parameter Design and Tolerance Design". *Universidad de Minnesota*.
8. **Seifi, A. Ponnambalam, K. Vlach, J.** (1999). "A Unified Approach to Statistical Design Centering of Integrated Circuits with Correlated Parameters". *Life fellow, IEEE*.
9. **Ngoi, B.K.A Ong, C.T.** (1996). "Optimum Assembly Using a Component Dimensioning Method". *Advanced Manufacturing Technology*.
10. **C. C. Yang and V.N.A. Naikan**, "Optimum Design of Component Tolerances of Assemblies Using Constraint Networks," *International Journal of Production Economics*, vol.84, no.2, May, 2003, pp.149-163
11. Weill, R. "Integrating Dimensioning and Tolerancing in Computer Aided Process Planning" *Robotics and computer Integrated Manufacturing* (Pergamon Press), Vol. 4, Num. 1-2, 1988, 41-48

ENLACES CONSULTADOS.

1. Análisis Multivariante.

<http://halweb.uc3m.es> .

Diplomatura en Estadística. Escuela Politécnica Superior.

Universidad Carlos III de Madrid.

2. Análisis estadístico de tolerancias.

<http://www.unizar.es/aeipro/finder/INGENIERIA%20DE%20PRODUCTOS/BB02.htm>)

José M^a Gurruchaga Vázquez.

Escuela Superior de Ingenieros de San Sebastián.

Universidad de Navarra.

3. Nuevos métodos de análisis multivariante.

<http://www.ub.edu/stat/personal/cuadras/metodos.pdf>.

Departamento de estadística.

Universidad de Barcelona.

4. Association for the Development of Computer-Aided Tolerancing Systems.

<http://adcats.et.byu.edu/home.php> .

5. Dimensional Control Systems.

www.3dcs.com .

6. American Society of Mechanical Engineers (ASME).

www.asme.org.

ANEXOS

ANEXO A: ARTÍCULOS.

[A.1]. Statistical tolerance synthesis with correlated variables.

Isabel González - Ismael Sánchez.

Universidad Carlos III de Madrid. (2008).

[A.2]. Tolerance desing optimization of machine elements using genetic algorithm.

A. Noorul Haq - k.Sivakumar - R.Saravanan - V. Muthiah.

Advanced Manufacturing Technology (2004).

[A.3]. Optimal Design Centring Through a Hybrid Approach Base don Evolutionary Algorithms and Monte Carlo Simulation.

Pierluissi, L. - Claudio, M. - Rocco, S.

Universidad central de Venezuela.

[A.4]. Método de la secante.

Ing Yamil Armando Cerquera Rojas

Docente Universidad Surcolombiana

www.faqmania.com/.../yacerque_200709152918_42068100_secante.pdf

[A.5]. Método de bisección.

Ángeles Carmona

Universidad politécnica de Cataluña

www-ma3.upc.edu/users/carmona/teaching/.../metodo%20biseccion.pdf

ANEXO B: PROGRAMACIÓN EN MATLAB.

[B.1]. Asignación de tolerancias y valores nominales en una pieza:

DepositoEtapa1.
DepositoEtapa1AG.
DepositoEtapa2.

[B.2]. Asignación de tolerancias y valores nominales en un ensamble de Bloques:

BloquesEtapa1.
BloquesEtapa1AG.
BloquesEtapa2.

[B.3]. Asignación de tolerancias y valores nominales en un ensamble de cilindros:

CilindrosEtapa1.
CilindrosEtapa1AG.
CilindrosEtapa2.

Optimal design of tolerances with correlated variables

Isabel González*

Department of Mechanical Engineering
Universidad Carlos III de Madrid

Ismael Sánchez†

Department of Statistics
Universidad Carlos III de Madrid

February 25, 2008

Abstract

Optimal tolerance design aims at assigning tolerances such that the functionality requirements are achieved with minimum cost. Classical tolerance design procedures are based on the assumption of independence of variables. This assumption might not be realistic, leading to the assign of non-optimal tolerances. This paper proposes an innovative methodology to allocate optimal statistical tolerances to dependent variables, where the dependence structure is estimated from the manufacturing process. The methodology is based on the assumption that the multivariate process variability is consequence of a set of independent factors. Hence, the tolerance assignment should be based on the statistical properties of these factors. The tolerance design takes the independence of the factors as a restriction, and tolerances are optimally assigned according to the variability of them.

Keywords: Tolerance Synthesis, Tolerance Allocation, Correlated variables

1 Introduction

The quality and functionality of parts are usually related with one or more variables $\mathbb{Y} = (Y_1, Y_2, \dots, Y_J)'$, such us length, width, weight, voltage, and so on. The functionality of the part imposes some specification limits on \mathbb{Y} . These variables can, in turn, be a function of other variables $\mathbb{X} = (X_1, X_2, \dots, X_K)'$.

*Isabel González. Universidad Carlos III de Madrid. Avd. de la Universidad 30, 28911, Leganés, Madrid (Spain), imgfariar@ing.uc3m.es, phone: 34-916245860, fax: 34-916249430

†Correspondence author: Ismael Sánchez. Universidad Carlos III de Madrid. Avd. de la Universidad 30, 28911, Leganés, Madrid (Spain), ismael@est-econ.uc3m.es, phone: 34-916249179, fax: 34-916249430. <http://halweb.uc3m.es/ismael>.

ERROR: undefinedresource
OFFENDING COMMAND: findresource

STACK:

/0
/CSA
/0
/CSA
-mark-