

Universidad de Matanzas “Camilo Cienfuegos”.
Facultad de Ingenierías Química y Mecánica.



IMPLEMENTACIÓN DE LA OPTIMIZACIÓN MULTIOBJETIVO DEL PROCESO DE TORNEADO UTILIZANDO ALGORITMOS GENÉTICOS.

Tesis Presentada en Opción al Título Académico de
Master en Ingeniería Asistida por Computadora.

Autor: Ing. Elvio Riquenes Galán.

Tutor: Dr.C. Ramón Quiza Sardiñas.

Matanzas, 2010.

*La vida es una obra de teatro que no permite ensayos...
Por eso, canta, ríe, baila, llora y vive intensamente
cada momento de tu vida...
... antes que el telón baje
y la obra termine sin aplausos.*

Charles Chaplin.

AGRADECIMIENTOS.

Al final del camino son muchas las cosas por agradecer, así como muchas las personas a quiénes agradecer. Con el temor de no equivocarme en olvidos, ni en posición de recuerdo, detalle que todos miran y pocos perdonan, prefiero agradecerles sin nombres, ni orden, ni preferencia.

En este sentido quiero agradecer a mis padres y hermanos por confiar en mí y apoyarme, a mi familia por existir, en especial a un viejito que mira por mis ojos, escucha por mis oídos y me apoya incondicionalmente en todo lo que decido, por sus enseñanzas, cariño y amor, a mi tutor que a pesar de estar al punto de volverme loco siempre estuvo luchando conmigo para que saliera el fruto de tanto esfuerzo, a otros profesores que me ayudaron en un momento determinado desde mis inicios, al Harry Potter mecánico que ha estado acompañándome desde los primeros días de la carrera hasta la actualidad, colaborando en todo momento y todos los sentidos, además le toca un por ciento del merito de mi título de ingeniero y ahora de este que estoy optando, a una abogada que tiene un cable a tierra pero siempre pendiente de mi como su brother mas chiquito, dándome apoyo y regaños cuando los he merecidos para lograr esto y muchas cosas más en mi vida, a Gladis que cuando mas necesite de un hombro amigo, allí estaba sin pensarlo y sin reparos, que me ha demostrado ser amigo y hermano, a mi hermano

jimagua de cerebro que siempre está presente a la hora clave aportando sus conocimientos, consejos y apoyo en la vida tanto profesional como no profesional, a dos chicas que a pesar de estar a cientos de millas siempre han estado pendiente de mis cosas, dándome el cariño y apoyo necesario para seguir adelante con mis planes futuros, confiando en que si puedo esto y mucho más, a una pelusa especial y bella (mi niña psicóloga) a la cual quiero mucho, ella lo sabe pero no lo quiere entender, por estar a mi lado, en mi mente y mi corazón, por apoyarme tanto en mi vida profesional como en mi vida en general, por enseñarme a ver la vida desde otro punto de vista, por darme fuerzas para continuar con lo iniciado y no dejarlo a medias, por soportar todas mis pesadeces y malacrianzas, por encontrar en mi cosas que otras personas no ven ó no tienen en cuenta, en fin por llegar a mi vida y ojala sea por mucho - mucho tiempo. A una ingeniera informática con alma de niña que le encantan los muñes asiáticos, por prestarme atención, ayuda y brindarme su tiempo y su tecnología para trabajar, primero en mi carrera de ingeniería y ahora en la maestría, además por ser una amiga incondicional, a mi sangre, mi amigo, hermano y brazo derecho en la informática que ha luchado incansablemente por lo que quiere hasta lograrlo, por apoyarme desde que curse mi carrera hasta esta meta que me propuse alcanzar, dedicando tiempo del poco que le quedaba libre de sus estudios para mí, tanto en la UCI como en Venezuela pero buscando algunos minutos y cumplir con mis exigencias, a un excelente deportista

de teoría y práctica que está en la misma contienda conmigo y ambos nos hemos servido de apoyo para seguir adelante, además a su madre que ha luchado contra todo por su vida, que me ha acogido como su hijo y no ha dejado que me rinda y no cumpla mis metas, además por acogerme entre su familia de amigos, a un gordo camionero que para él soy su hijo, su hermano y su amigo que siempre está presente y pendiente de mi para lo que lo necesite, impulsándome y apresurándome para que termine lo que inicie y no lo deje a medias, a un compañero y amigo que a veces desayuna con batido de limón y que ha tenido paciencia conmigo, que me ha brindado su apoyo y el tiempo que he necesitado para mis cursos en la universidad, trabajos y por último la tesis, que a veces es insoportable y a pesar de tener nuestras diferencias me acogió como su amigo, a un loco que siempre me molesta con las ordenes de trabajo, a ese amigo que ha tenido paciencia cada vez que necesito la computadora y cuando ha esta ha estado rota siempre ha buscado alternativas para que avance en mis cosas de la maestría, además siempre pendiente de cuanto me hace falta, tal vez él desee más que yo que termine mi tesis y acabe de graduarme. Al que bautice con el nombre del Yeti que desde un principio no tuvo reparos en decirme que si cuando le pedí autorización para continuar con mi maestría, al cual le debo muchas cosas que he aprendido en vida profesional, que siempre confió en mí y nunca dudo de mi trabajo, gracias por ser mi conmigo hombre y amigo antes de ser jefe, a mi consejera telefónica, que nos

servimos de consuelo en muchas ocasiones, que me enseñó que en la vida hay que echar pa'lante, que cuando uno se da un golpe en para crecer y que de todo lo malo se saca alguna enseñanza buena, que siempre hay uno más mal que tú, que no me deja ni me perdona que me rinda por nada ni por nadie, que siempre está disponible para cuando la necesito, escucha todas mis muelas sin objeción y me aconseja como si fuera mi madre, a los chicos del piquete de la pachanga que nos inventamos cualquier cumpleaños o días festivos para despejar de todos los problemas. A un amigo y hermano hondureño que aunque actualmente este lejos físicamente siempre ha estado apoyándome y ayudándome cada vez que lo he necesitado, que se preocupa por lo que me suceda, que tiene un porciento de influencia en todo el tiempo de mi carrera profesional, a todos mis hermanos de corazón que siempre puedo contar con ellos, a todo aquel que de cierta forma creyó y pensó que yo no podía, eso me dio fuerzas para demostrarles que yo si puedo y aquí está el resultado yo pude más que esos mediocres, en fin a todo aquel que quiere el bien para mí.

DECLARACIÓN DE AUTORIDAD.

Declaro que soy el único autor de este trabajo, que pertenece íntegramente a la Facultad de Ingenierías Química y Mecánica de la Universidad de Matanzas. Autorizo su consulta a otras instituciones, a los profesionales, técnicos y personas en general que lo necesiten, siempre que se respete la procedencia del mismo, quedando prohibida la reproducción total o parcial de este documento, sin la autorización expresa de la Universidad de Matanzas.

Elvio Riquenes Galán.

RESUMEN.

La optimización multiobjetivo permite seleccionar los regímenes de corte óptimos en procesos de torneado, de una forma más efectiva y flexible. Además, los algoritmos genéticos han probado ser muy convenientes para este propósito. Este trabajo describe la implementación de una librería para MATLAB para permitir la aplicación de una técnica de optimización multi-objetivo, basada en algoritmos genéticos, y desde un enfoque *a posteriori*, para procesos de torneado. En la tesis, los algoritmos empleados y los códigos implementados son explicados, con el objetivo de servir como manual de usuario de la librería. Dos casos de estudio, uno de ellos para torneado de una sola pasada y el otro para torneado de pasadas múltiples, fueron ejecutados para verificar el correcto funcionamiento de la librería, y sus respectivos resultados fueron analizados.

ABSTRACT.

Multi-objective optimization allows selecting optimal cutting parameters in turning processes, in a more effective flexible way. Moreover, genetic algorithms have been probed being very convenient for this purpose. This work describes the implementation of a toolbox for MATLAB to allow application of a genetic algorithm – based, multi-objective optimization technique for turning processes, by using an *a posteriori* approach. In the thesis employed algorithms and implemented codes are explained, in order to be used as user’s guide of the toolbox. Two cases, one of them for a single-pass turning process and the other one for a multi-pass turning process, were executed to verify the right functioning of the toolbox, and their respective outcomes were analyzed.

TABLA DE CONTENIDO.

INTRODUCCIÓN.....	1
CAPÍTULO 1. Estado del Arte.....	5
1.1 Principios de la optimización del proceso de torneado.	5
1.1.1 Bases conceptuales.	5
1.1.2 Criterios de optimización.	7
1.1.3 Restricciones.....	12
1.2 Optimización de los procesos de corte.	14
1.3 Definición del problema de optimización multiobjetivo.	16
1.4 Técnicas de optimización.	20
1.4.1 Técnicas numéricas.....	20
1.4.2 Técnicas de búsqueda heurística.....	24
CAPÍTULO 2. Desarrollo de la Aplicación.	30
2.1 Definición de la metodología propuesta.	30
2.1.1 Variables de decisión.....	30
2.1.2 Función objetivo.....	31
2.1.3 Restricciones.....	32
2.1.4 Algoritmo genético.....	33
2.2 Implementación de la metodología propuesta.	36
2.2.1 Función creapop.	36
2.2.2 Función gaenc.....	38
2.2.3 Función gadec.....	39

2.2.4 Función defpop.	40
2.2.5 Función evalpop.....	42
2.2.6 Función selturn.	43
2.2.7 Función crossover.	46
2.2.8 Función mutate.	47
2.2.9 Función parpop.....	49
2.2.10 Función filtpop.....	50
CAPÍTULO 3. Estudios de Caso.	54
3.1 Caso de estudio 1: Torneado de una sola pasada.	54
3.1.1 Definición del caso de estudio.	54
3.1.2 Implementación del problema.....	55
3.1.3 Análisis de los resultados.	58
3.2 Caso de estudio 2: Torneado de pasadas múltiples.....	60
3.2.1 Definición del caso de estudio.	60
3.2.2 Implementación del problema.....	61
3.2.3 Análisis de los resultados.	65
3.3. Conclusiones parciales del capítulo.....	67
CONCLUSIONES.	68
RECOMENDACIONES.	69
BIBLIOGRAFÍA.	70

INTRODUCCIÓN.

En la actualidad, la selección de tecnologías de torneado se basan en valores de tanteo, recomendados por tablas y manuales (Dereli *et al.*, 2001). Esto se debe a la complejidad de los procesos de corte que hacen muy difíciles las labores de optimización (Kumar y Kumar, 2000). Además, existen otras causas que afectan el éxito de la optimización de los procesos de maquinado (Sönmez *et al.*, 1999), entre las que se encuentran: la necesidad de contar con modelos para las variables del proceso (vida útil de la herramienta, fuerza de corte, etc.), que permitan establecer relaciones y restricciones cercanas a la realidad; la complejidad de las características de la máquina y de la herramienta de corte, que muchas veces incluyen variables cualitativas o discontinuas; la dificultad en la selección de un criterio efectivo de optimización; y las limitaciones de las técnicas de optimización matemáticas y numéricas.

Todo lo anterior ha provocado que el desarrollo de los métodos de optimización del maquinado, a pesar de la cantidad de investigaciones que le han sido dedicadas, haya sido sorprendentemente lento (Wang *et al.*, 2002) y, la mayor parte de ellas, orientada a la optimización de un sólo objetivo, bien sea el costo de producción (Liang *et al.*, 2001; Wang *et al.*, 2002; Saravanamn *et al.*, 2003; Cus y Balic, 2003), el tiempo de elaboración (Chua *et al.*, 1991; Ahmad y Anwarul, 2001; Quiza y Rivas,

2003) o la velocidad de remoción de material (Ko y Kim, 1998; Chien y Tsai, 2003).

Esta consideración de un sólo objetivo ofrece una visión simplificada del problema, por lo cual tiende a ser reemplazada por la optimización de criterios múltiples bien desde un enfoque a priori (Zuperl y Cus, 2003; Cus y Balic, 2003) o bien, a posteriori (Quiza *et al.*, 2005; Özel, 2009). No obstante, los trabajos anteriores se dirigen a la optimización de regímenes de corte para una tecnología de fabricación preestablecida, lo cual conduce óptimos locales para cada paso tecnológico, cuya combinación no necesariamente coincide con el óptimo global.

Por lo tanto, se está en presencia del siguiente **problema científico**: los métodos de optimización reportados en la literatura especializada, no permiten obtener los valores óptimos, considerando criterios múltiples, en el diseño de tecnologías de maquinado.

Lo anterior, como se pudo ver, está provocado por la alta complejidad del proceso modelado. Por otro lado, en los últimos años, se ha incrementado el uso de algoritmos evolutivos en los problemas optimización (Kurpati *et al.*, 2002). Dentro de estas técnicas, los algoritmos genéticos son la estrella indiscutible (Schneider y Kirkpatrick, 2006). A diferencia de otros métodos de optimización, los algoritmos genéticos son muy poco exigentes con las funciones que emplean. Consideraciones de convexidad o concavidad, derivabilidad y continuidad

no son en absoluto necesarias para las funciones a optimizar mediante algoritmos genéticos (Abbass *et al.*, 2001). En especial, los algoritmos genéticos son muy útiles en la optimización multiobjetivo, ya que permiten obtener, simultáneamente, un conjunto de puntos del espacio de búsqueda que definan completamente la frontera de Pareto (Schneider y Kirkpatrick, 2006).

Por lo tanto, como **hipótesis** de este trabajo se propone que: mediante el uso de algoritmos genéticos, es posible llevar a cabo la optimización de la tecnología de torneado de piezas de tipo de árboles y ejes, según criterios múltiples y siguiendo un enfoque a posteriori.

Para validar la hipótesis anterior, como **objetivo general** del presente trabajo se plantea: desarrollar e implementar un método de optimización multiobjetivos del diseño de tecnologías de torneado de piezas de tipo árboles y ejes, basado en el uso de algoritmos genéticos.

Con el propósito de llevar a cabo el objetivo general, se han trazado los siguientes **objetivos específicos**:

1. Definir la metodología de optimización a emplear.
2. Establecer la estructura de la aplicación, a través de la determinación y descripción de las subrutinas que la componen.
3. Establecer los algoritmos de cada subrutina de la aplicación.
4. Implementar cada subrutina de la aplicación.

5. Validar la aplicación desarrollada con varios estudios de caso.

CAPÍTULO 1. ESTADO DEL ARTE.

Este capítulo tiene como objetivo establecer los presupuestos teóricos fundamentales de la investigación desarrollada, basándose en un análisis crítico de las fuentes bibliográficas actualizadas, consultadas al respecto.

1.1 Principios de la optimización del proceso de torneado.

1.1.1 Bases conceptuales.

La optimización de los parámetros de corte es un componente importante en la planificación de los procesos de maquinado. Sin embargo, el análisis profundo del proceso de corte de metales implica ciertos costos, particularmente en el caso de series pequeñas. En el caso de maquinado individual, es particularmente necesario que se acorte tanto como sea posible el procedimiento para la determinación de los parámetros de cortes óptimos, pues el análisis del costo podría exceder la eficacia económica que se alcanzaría si se trabajara con las condiciones óptimas.

Para la optimización de los parámetros de la máquina, los métodos cuantitativos se han desarrollado principalmente con la consideración de un solo objetivo, como la minimización de costo o maximización de ganancia, etc. Para el proceso de un solo objetivo de optimización, se han propuesto varias técnicas diferentes, como el cálculo diferencial, análisis

de regresión, la programación lineal, la programación geométrica y estocástica, y la simulación mediante computadoras.

Lamentablemente, en muchas aplicaciones reales, los fabricantes enfrentan problema de optimización simultánea de varios objetivos. Esos objetivos son a menudo contradictorios e incomparables.

La optimización de los procesos productivos, en general, se puede llevar a cabo en tres niveles básicos (Shaw, 1984):

1. Establecimiento de mejoras relativas a los actuales métodos de producción.
2. Selección de nuevos métodos de producción con el equipamiento existente.
3. Desarrollo de equipamiento nuevo.

Evidentemente, los niveles 2 y 3 requieren de un esfuerzo y una inversión normalmente grande, por lo que el trabajo del tecnólogo comúnmente se limita al nivel 1, o sea, a la determinación de los parámetros óptimos dentro de un proceso establecido.

En el torneado, como en el resto del maquinado, los parámetros principales que controlan el proceso, son las variables del régimen de corte: profundidad, avance y revoluciones del husillo. Por tanto, la optimización del mismo se basa en la selección de un régimen de corte que maximice o minimice (según corresponda) uno o varios criterios determinados.

Dada la complejidad del proceso de corte, las tareas de optimización se hacen muy difíciles (Kumar y Kumar, 2000). Además, existen otras causas que afectan el éxito de la optimización del torneado (Sönmez *et al.*, 1999), entre las que se encuentran: la necesidad de contar con modelos para las variables del proceso (la vida útil de la herramienta, la fuerza de corte, etc.), que permitan establecer relaciones y restricciones cercanas a la realidad; la complejidad de las características de la máquina y de la herramienta de corte, que muchas veces incluyen variables cualitativas o discontinuas; la dificultad en la selección de un criterio efectivo de optimización; y las limitaciones de las técnicas de optimización matemáticas y numéricas.

Todo lo anterior ha provocado que el desarrollo de los métodos de optimización de torneado (y de maquinado, en general), a pesar de la cantidad de investigaciones que le han sido dedicadas, haya sido sorprendentemente lento (Wang *et al.*, 2002).

1.1.2 Criterios de optimización.

La selección del criterio de optimización (o sea, de la función objetivo a optimizar), es, quizás, el punto más importante en cualquier estudio sobre el tema. Tradicionalmente, el criterio utilizado, se ha basado en el costo de la producción. En este sentido se enfocan desde el primer trabajo conocido sobre optimización (Taylor, 1907), hasta investigaciones muy recientes (Liang *et al.*, 2001; Wang *et al.*, 2002; Saravanamn *et al.*,

2003; Cus y Balic, 2003; Singh *et al.* 2006; Yildiz y Ozturk, 2006; Paiva *et al.*, 2007; Kim *et al.* 2008).

La expresión de costo utilizada varía ligeramente de un autor a otro. Shaw (1984) considera el costo por pieza, Z , como la suma de tres costos básicos: costo directo de la máquina y el trabajo, Z_0 , el costo por cambio de la herramienta, Z_C y el costo de herramienta para cada pieza Z_T :

$$Z = Z_0 + Z_C + Z_T = z_0\tau + z_0\tau_C \frac{\tau}{T} + z_T \frac{\tau}{T} \quad (1.1)$$

donde z_0 es el costo de máquina y salario por unidad de tiempo; z_T , el costo de cada filo de la herramienta; τ , el tiempo de maquinado y τ_C , el tiempo invertido en el cambio de la herramienta. Esta misma expresión utilizan Kayakan y coautores (1996); Meng y coautores (2000); Amiolemhen e Ibhádode (2004) y Saravanan y coautores (2003), aunque estos últimos introducen una pequeña modificación sin trascendencia.

Bhaskara y colaboradores (1998) minimizan el costo total de elaboración en operaciones de pasadas múltiples, mediante la minimización del costo de cada pasada.

La principal limitante de esta expresión es que considera que el tiempo de trabajo de la máquina es igual al tiempo de maquinado, lo cual es cierto sólo para operaciones de una pasada. Para solucionar esta dificultad, Wang *et al.* (2002) diferencian entre el tiempo de maquinado, τ , y tiempo de trabajo de la máquina, τ' , resultado la expresión:

$$Z = Z_0(\tau_s + \tau' + \tau_c \frac{\tau}{T}) + z_T \frac{\tau}{T} \quad (1.2)$$

donde τ_s es el tiempo preparativo y conclusivo. Si bien la efectividad de este criterio, desde el punto de vista económico es indiscutible, el enfoque que se le da en la literatura consultada, no es completamente satisfactorio, según criterio del autor de esta tesis. Esto se debe a que las expresiones para el cálculo del costo de cada pieza elaborada, están deducidas considerando un flujo continuo en la producción, lo cual no siempre corresponde a la práctica industrial real.

Otro criterio de optimización que comúnmente se utiliza, es el tiempo de elaboración (Chua et al., 1991; Ahmad y Anwarul, 2001; Quiza y Rivas, 2003; Basak *et al.*, 2007), que se basa en el presupuesto de que el costo de herramienta es despreciable frente al monto total (Sandvik, 1994). El tiempo de elaboración se asume igual o proporcional al tiempo de maquinado:

$$\tau = \frac{L + L' + L''}{f \cdot n} \quad (1.3)$$

donde L es la longitud a elaborar; L' y L'' las distancias recorrida por la herramienta en la entrada y la salida del corte, respectivamente; f , el avance; y n , la velocidad de rotación del husillo. Como se puede apreciar, la expresión anterior desprecia el tiempo de cambio de la herramienta, y el de trabajo en vacío.

Algunas veces, se emplea como función objetivo la cantidad de metal cortado por unidad de tiempo o velocidad de remoción de material,

M (*material removal rate*), que no es más que otra forma de considerar el tiempo de elaboración. Entre los trabajos que utilizan este criterio están el de Chien y Tsai (2003) y el de Ko y Kim (1998). Este parámetro puede calcularse, para el torneado, como:

$$M = 10^3 \cdot v \cdot f \cdot a \quad (1.4)$$

Además, si bien la consideración de que el costo de herramienta es despreciable, está justificado en las condiciones de la industria de los países desarrollados, para los países en vías de desarrollo deja de tener validez absoluta (Quiza *et al.*, 2006).

Algunos autores ofrecen modelos que permiten considerar varios objetivos, aunque no de forma simultánea. En este grupo se encuentran el de Cakir y Gurarda (1998) y el de Lee y Tarng (2000) que posibilitan la optimización bien del costo de producción o bien de la cantidad de metal cortado producción, o el de Basak *et al.* (2007) que optimiza el costo de producción o el tiempo de elaboración.

Una solución a la sobrevaloración de uno de los criterios de optimización es emplear múltiples objetivos. Esta es una tendencia que gana terreno en la actualidad (Zuperl y Cus, 2003; Cus y Balic, 2003). No obstante, el enfoque de optimización que a predominado a sido *a priori*, que se basa en la integración de los diversos objetivos en una única función, generalmente mediante una adecuada ponderación ofrecida por expertos. En este sentido están orientados los trabajos de Cus y Balic (2003), de Zuperl y Cus (2003) y de Cus y Zuperl (2006), que utilizan una

denominada “función implícita del fabricante”, que integra tres objetivos independientes: tiempo de elaboración, costo de producción y calidad.

El autor de esta tesis considera que este enfoque tiene una limitación, dada por el requerimiento de una valoración previa, casi siempre subjetiva, de la importancia relativa de cada objetivo de optimización.

Croitoru, Severincu y Belous (2001) han desarrollado un método de optimización multiobjetivo, pero éste no se adapta a variables discontinuas y, además, el método de solución es gráfico, lo que hace muy lento y engorroso el proceso de cálculo.

Tres casos de estudio muy interesantes de optimización multiobjetivo de torneado de acero endurecido son presentados por Özel (2009), en los cuales se emplean como criterios de optimización la rugosidad superficial y el tiempo de maquinado (caso 1), vida útil de la herramienta y velocidad de remoción del material (caso 2) y tensiones residuales y rugosidad superficial (caso 3).

Una alternativa es la optimización multiobjetivo, considerando dos criterios que deben ser simultáneamente minimizados: el tiempo de elaboración y el gasto de herramienta, lo cual parece ser más adecuado a las condiciones de la industria cubana. Este enfoque ha sido aplicado, exitosamente al torneado tanto de una sola pasada (Quiza *et al.*, 2006) como de pasadas múltiples (Quiza *et al.*, 2009).

1.1.3 Restricciones.

Un aspecto muy importante dentro de toda optimización es establecer las restricciones adecuadas, las cuales dependen de las características del proceso. En el torneado, se reporta el uso de varias restricciones que garantizan la factibilidad del régimen seleccionado, en dependencia de las condiciones específicas: máquina, herramienta, material elaborado, etc.

Entre las restricciones utilizadas por casi todos los autores (Chua *et al.*, 1991; Kayakan *et al.*, 1996; Cakir y Gurarda, 1998; Lee y Tarng, 2000; Ahmad y Anwarul, 2001; Wang *et al.*, 2002; Saravanan *et al.*, 2003; Quiza y Rivas, 2003; Quiza y Rivas, 2004; Paiva *et al.*, 2007; Muthukrishnan y David, 2009) están mantener los parámetros del régimen de corte (profundidad, avance y velocidad de corte) dentro de los rangos admitidos por la herramienta:

$$a_{\text{MIN}} \leq a \leq a_{\text{MAX}}, \quad (1.5a)$$

$$f_{\text{MIN}} \leq f \leq f_{\text{MAX}}, \quad (1.5b)$$

$$v_{\text{MIN}} \leq v \leq v_{\text{MAX}}. \quad (1.5c)$$

Otra restricción que hay que tener en cuenta, en pasadas de desbaste, es que la fuerza (F_c) y la potencia de corte (P_c) no superen los valores máximos admisible por la máquina herramienta (Chua *et al.*, 1991; Cakir y Gurarda, 1998; Ko y Kim, 1998; Saravanan *et al.*, 2003; Kayakan *et al.*, 1996; Meng, Arsecularatne y Mathew, 2000; Lee y Tarng,

2000; Ahmad y Anwarul, 2001; Wang et al., 2002; Quiza y Rivas, 2003; Quiza y Rivas, 2004; Yildiz y Ozturk, 2006; Kim *et al.*, 2008):

$$F_C \leq F_{MAX}. \quad (1.6a)$$

$$P_C \leq P_{MAX}. \quad (1.6b)$$

Para pasadas de acabado, es importante considerar que la rugosidad superficial (R) no supere a la establecida en los requisitos técnicos de la superficie elaborada (Cakir y Gurarda, 1998; Ko y Kim, 1998; Lee y Tarng, 2000; Saravanan *et al.*, 2003; Wang et al., 2002):

$$R \leq R_{MAX}. \quad (1.7)$$

Una restricción adicional propuesta en algunos trabajos (Saravanan *et al.*, 2003; Singh *et al.*, 2006; Yildiz y Ozturk, 2006; Kim *et al.*, 2008), que presenta cierto interés, es que la temperatura en la zona de corte, θ , no supere cierto valor admisible (θ_{MAX}). Sin embargo en ninguno de los trabajos citados no se señala como seleccionar esta temperatura máxima. Alfonso (2002), para el torneado de aceros templados, plantea que esta temperatura debe ser la de recristalización del material elaborado. Alfonso propone utilizar, en lugar de la temperatura, la termo-fem, ε , que no debe superar aquella a la cual ocurre la recristalización.

Adicionalmente, Nefiodov y Osipov (1980), plantean que la herramienta utilizada, bajo las condiciones de trabajo seleccionadas, deben cumplir las condiciones de resistencia y rigidez.

$$F_C \leq \frac{BH^2\sigma_F}{6L_{VOL}}. \quad (1.8a)$$

$$F_C \leq \frac{y_{MAX}EBH^3}{4L_{VOL}}. \quad (1.8b)$$

donde F_C es la componente tangencial de la fuerza de corte; B y H , el ancho y la altura de la sección transversal del vástago de la herramienta; σ_F , la tensión límite a la flexión del material del vástago; E , el módulo de elasticidad del material del vástago; L_{VOL} , la longitud del voladizo de la herramienta; y y_{MAX} , la flecha de pandeo máxima admisible para la punta de la herramienta (0,1 mm para el torneado y 0,05 para el de acabado).

Finalmente, Singh y coautores (2006), así como Yildiz y Ozturk (2006) consideran como restricción que el régimen de corte sea seleccionado dentro de la zona que garantiza la estabilidad del corte.

1.2 Optimización de los procesos de corte.

Para la optimización de regímenes de corte en el torneado, se han empleado diversos métodos. El primero, muy difundido por su relativa sencillez y su claro significado matemático, es el analítico, que se basa en el uso de las derivadas de la función objetivo. Utilizado por primera vez por Taylor (1907) para determinar la velocidad de corte óptima en operaciones de una sola pasada, este método es ampliamente descrito en libros y manuales (Shaw, 1984; Sandvik, 1994) y aún se mantiene en uso (Kayakan *et al.*, 1996; Wang *et al.*, 2002).

La principal limitación del método analítico es su dificultad al aplicarlo al régimen de corte, donde la función objetivo no siempre es derivable y, frecuentemente, incluye variables discretas o discontinuas. Aunque se han hecho intentos para solucionar esta dificultad (Shaw, 1984), los resultados conducen a expresiones muy complejas y de validez muy discutible.

La programación dinámica y la programación geométrica también han sido aplicadas a la optimización de regímenes de corte (Chua *et al.*, 1991; Cakir y Gurarda, 1998; Lee y Tarng, 2000). Estas técnicas ofrecen buenos resultados en sistemas no lineales, pero cuando el número de parámetros involucrados se hace grande, el esfuerzo computacional que requieren las hace impracticables.

También el método robusto de Tagushi (Nian *et al.*, 1999; Aslan *et al.*, 2007; Hasçalik y Çaydas, 2007) y el algoritmo de hormiguero (*ant colony algorithm*) (Vijayakumar *et al.*, 2003) han sido empleado para la selección de regímenes de corte óptimos en el torneado, con cierto éxito.

En los últimos años, se ha incrementado el uso de algoritmos evolutivos en los problemas optimización (Kurpati *et al.*, 2002). Dentro de estas técnicas, los algoritmos genéticos (AG's) son la estrella indiscutible.

Varios autores han reportado el empleo de AG's para la selección de regímenes de corte óptimos en el torneado (Bhaskara *et al.*, 1998; Ahmad y Anwarul, 2001; Antonio y Davim, 2002; Cus y Balic, 2003; Quiza y Rivas, 2003; Amiolemhen y Ibhádode, 2004). Saravanan y

coautores (2003) proponen un método basado en AG's para la optimización del torneado, y comparan los resultados con los ofrecidos por un método basado en la cristalización simulada (*simulated annealing*), propuesto en el mismo artículo. La eficacia reportada para ambos métodos es similar.

Chien y Tsai (2003) y Yildiz y Ozturk (2006) no sólo emplean un AG para optimizar el régimen de corte, sino que seleccionan los parámetros más convenientes para el GA, mediante el método de Tagushi.

Cus y Zuperl (2006) utilizan un AG para optimizar varios criterios, simultáneamente, pero los agregan según una función lineal, por lo cual el enfoque es *a priori*.

Özel (2009) reporte el uso del método de optimización por enjambre de partículas (*particle swam optimization*) según un enfoque *a posteriori* del proceso de torneado de acero H13 endurecido, para diferentes objetivos, los cuales se combina de dos en dos.

Por su parte, los algoritmos genéticos han sido empleados por Quiza y colaboradores (2006, 2009) para la optimización multiobjetivo de procesos de taladrado tanto de una como de varias pasadas, también desde un enfoque *a posteriori*.

1.3 Definición del problema de optimización multiobjetivo.

La optimización mono-objetivo puede verse como el problema de determinar un vector de variables de decisión, \mathbf{x} , que satisfaga las

restricciones y optimice una función escalar, y . Aquí, el término «optimizar» significa encontrar un valor mínimo o máximo; sin embargo, es posible tratar sólo con problemas de minimización ya que cualquier problema de maximización puede transformarse en uno de minimización transformando convenientemente la función objetivo.

De un modo formal, un problema de optimización monoobjetivo puede definirse de la siguiente forma (Özel, 2009):

Definición 1.1 (*optimización monoobjetivo*): dada una función escalar $y : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}, \Omega \neq \emptyset$, encontrar el valor $\mathbf{x}^* \in \Omega$ (llamado solución óptima global), la cual minimiza (o maximiza) el valor de y , es decir, $\neg \exists \mathbf{x}' \in \Omega : y(\mathbf{x}') < y(\mathbf{x}^*)$.

El conjunto Ω es la región factible, la cual se define usualmente como:

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid (g_i(\mathbf{x}) \geq 0, i = 1, \dots, m) \wedge (h_i(\mathbf{x}) = 0, i = 1, \dots, p)\}$$

donde $g_i(\mathbf{x})$, son las m restricciones de desigualdad y $h_i(\mathbf{x})$, son las p restricciones de igualdad.

Por el contrario, la optimización multiobjetivo puede definirse como el problema de minimizar (o maximizar), simultáneamente, dos o más funciones objetivo. De un modo más formal:

Definición 1.2 (*optimización multiobjetivo*): Dada la función vectorial $\mathbf{y} : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^k, \Omega \neq \emptyset$, hallar el valor de $\mathbf{x} \in \Omega$ tal que minimice (o maximice) las componentes del vector \mathbf{y} .

Al igual que en un problemas de optimización mono-objetivo, Ω está limitado por restricciones de desigualdad y de igualdad.

Sin embargo, debido a la inexistencia de un criterio de comparación entre dos vectores, puede no existir un mínimo global para la optimización multiobjetivo. En el enfoque *a posteriori*, la toma de decisiones se toma luego del proceso de optimización, a partir de un conjunto de soluciones candidatas, óptimas en el sentido de que no hay ninguna otra dentro del espacio de búsqueda, es superior a ellas cuando se consideran todos los objetivos de optimización.

Para formalizar, por lo tanto, el enfoque a posteriori, es necesario realizar varias definiciones preliminares.

Definición 1.3 (*dominancia de Pareto*): Se dice que un vector $\mathbf{u} = \{u_1, \dots, u_k\}$ domina a otro vector $\mathbf{v} = \{v_1, \dots, v_k\}$ (lo cual se denota como $\mathbf{u} \preceq \mathbf{v}$), si y sólo si ninguna componente de \mathbf{u} es mayor que la componente respectiva de \mathbf{v} , y al menos una componente de \mathbf{u} es estrictamente menor; o sea:

$$\mathbf{u} \preceq \mathbf{v} : (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^k \Leftrightarrow (\forall i \in \{1, \dots, k\}, u_i \leq v_i) \wedge (\exists i \in \{1, \dots, k\} : u_i < v_i).$$

Definición 1.4 (*optimalidad de Pareto*): Se dice que una solución, $\mathbf{x} \in \Omega$, es un óptimo de Pareto con respecto a Ω , (se denota $\mathbf{x} = \text{par}(\Omega)$) si y sólo si no existe ningún $\mathbf{x}' \in \Omega$ para el cual $\mathbf{y}(\mathbf{x}') \preceq \mathbf{y}(\mathbf{x})$.

Las soluciones óptimas de Pareto también s denominan como soluciones no inferiores, admisibles o eficientes.

Definición 1.5 (*Conjunto óptimo de Pareto*): Es conjunto óptimo de Pareto (se denota \mathcal{P}) se define como el conjunto de todas las soluciones óptimas de Pareto para un problema de optimización multiobjetivo dado, o sea:

$$\mathcal{P} \triangleq \{\mathbf{x} \in \Omega \mid \neg \exists \mathbf{x}' \in \Omega : \mathbf{y}(\mathbf{x}') \preceq \mathbf{y}(\mathbf{x})\}.$$

Definición 1.6 (*Frontera de Pareto*): La frontera de Pareto (se denota \mathcal{F}) se define como el conjunto de las imágenes $\mathbf{y}(\mathbf{x})$ para todos los valores del conjunto óptimo de Pareto, o sea:

$$\mathcal{F} \triangleq \{\mathbf{y} = \mathbf{y}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}\}.$$

En la Fig. 1.1 se muestra una representación gráfica del conjunto óptimo y de la frontera de Pareto para una función bidimensional de argumento bidimensional.

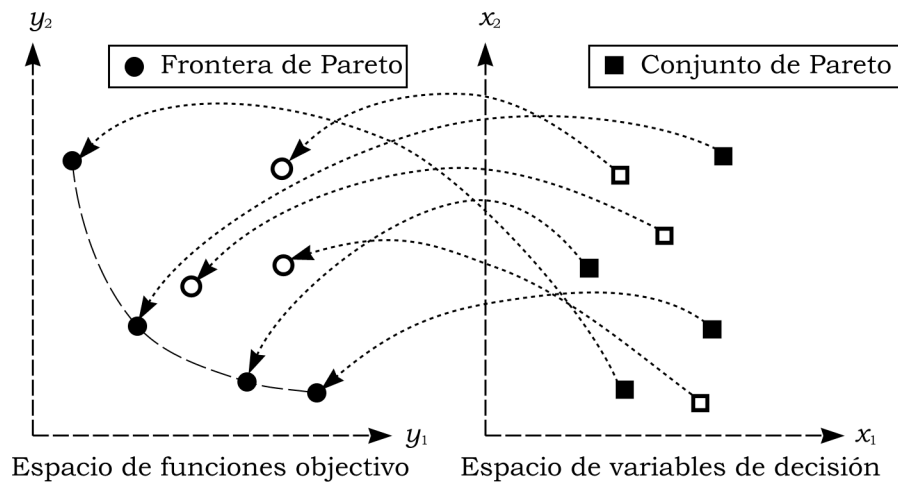


Figura 1.1 Representación gráfica del conjunto y la frontera de Pareto.

1.4 Técnicas de optimización.

1.4.1 Técnicas numéricas.

1.4.1.1 Metodología de superficie de respuesta.

La metodología de superficie de respuesta es un enfoque general para la obtención del valor óptimo de una variable dependiente (de respuesta) la cual es función de varias variables independientes (explicatorias). Esta técnica combina el diseño de experimentos y la regresión múltiple.

El diseño de experimentos es un enfoque general para el diseño de cualquier ejercicio de obtención de información donde tenga lugar variación. En la modelación de procesos de maquinado, el diseño experimental trata, fundamentalmente, con experimentos controlados, donde la variación en la variable dependiente está bajo el control del investigador.

Las técnicas actuales de diseño experimental se basan en los siguientes principios:

- ♦ comparación (ayuda a detectar variaciones no deseadas en los resultados medidos);
- ♦ aleatorización (disminuye la influencia de factores no controlados en los resultados medidos);
- ♦ replicación (permite estimar la variación en los resultados medidos);

- ♦ bloqueo (reduce las fuentes de variación, conocidas pero irrelevantes, entre las unidades, incrementando la precisión en las estimaciones de las fuentes de variación); y
- ♦ ortogonalidad (asegura que el contraste será llevado a cabo de modo tal que toda la información sea capturada).

El enfoque más común dentro del diseño de experimentos es el diseño factorial, el cual proporciona una distribución conveniente de los puntos experimentales, de acuerdo con los principios antemencionados. En la Fig. 1.2, se muestran varios ejemplos de diseño experimental para dos factores.

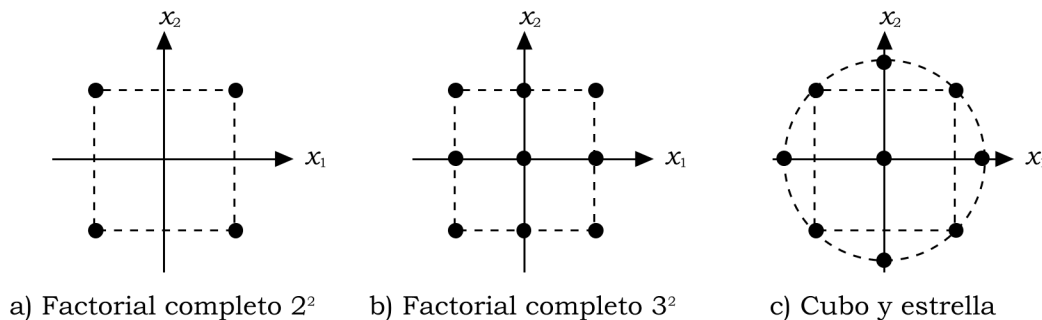


Figura 1.2 Ejemplos de diseño factorial (dos factores).

Diversos artículos han sido reportados con el uso de la metodología de superficie de respuesta al maquinado (Tang, 2006; Paiva *et al.*, 2006), tanto la metodología completa como algunos de sus componentes.

1.4.1.2 Método de Taguchi.

El método de Taguchi, también conocido como diseño robusto de Taguchi, es una metodología para diseñar productos o procesos que tengan una sensibilidad mínima a la variación de factores incontrolables.

Este método se basa en técnicas convencionales, pero con ciertos lineamientos específicos para el diseño experimental y en análisis de los resultados de los experimentos (Tong *et al.*, 2007).

Un aspecto clave en el método de Taguchi lo representa la llamada función de pérdida, la cual es una función continua definida en términos de la desviación del parámetro de diseño de su valor ideal (Jeang *et al.*, 2007).

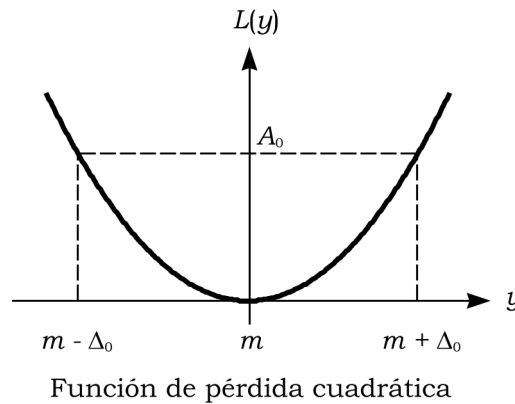


Figura 1.3 Función de pérdida cuadrática.

Esta función de pérdida $L(y)$ se asume, usualmente, cuadrática (Fig. 1.3), de forma que puede expresarse por la ecuación:

$$L(y) = k(y - m)^2; \tag{1.9}$$

donde m es el valor deseado para el parámetro de diseño y y su valor real, mientras que k es una constante que depende de los límites de especificación. Para un grupo de datos experimentales, con un valor medio μ y una varianza σ^2 , la función de pérdida promedio puede estimarse por:

$$L(y) = k[\sigma^2 + (\mu - m)^2]. \tag{1.10}$$

En el diseño de parámetros, hay dos tipos de factores que afectan la característica funcional del producto: factores de control y factores de ruido. Los primeros son aquellos que pueden ser controlados con facilidad; los segundos, por el contrario, son difíciles o imposibles de controlar.

Con el objetivo de variar y comprobar los diversos niveles de los factores de control, se utiliza una técnica especial de diseño experimental, llamada arreglos ortogonales. En realidad, se usan dos arreglos, que contienen los factores de control y de ruido. Ellos son denominados arreglos interior y exterior, respectivamente.

La combinación más conveniente de parámetros es calculada a partir de los datos experimentales usando la razón señal-ruido, (*signal-to-noise*, *SN*). Se consideran óptimos los niveles que maximizan la razón señal-ruido deseada. Existen tres razones estándares de señal-ruido, en dependencia del comportamiento de la respuesta deseada:

1. cuanto menor, mejor (para hacer la respuesta del sistema tan pequeña como sea posible):

$$SN = -10 \log \left(\frac{1}{n} \sum_i y_i^2 \right); \quad (1.11a)$$

2. nominal, mejor (para reducir la variabilidad alrededor de un valor deseado):

$$SN = 10 \log \left(\frac{\bar{y}^2}{\sigma^2} \right); \quad (1.11b)$$

3. cuanto mayor, mejor (para hacer la respuesta del sistema tan grande como sea posible):

$$SN = -10 \log \left(\frac{1}{n} \sum_i \frac{1}{y_i^2} \right). \quad (1.11c)$$

Una vez que la relación señal-ruido ha sido calculada para cada corrida experimental, se determina gráficamente los niveles factoriales que maximizan dicha relación, los cuales serían los niveles óptimos para el diseño en cuestión.

Se han reportado diversas aplicaciones del método de Taguchi, no sólo a la optimización de un solo objetivo (Xueping *et al.*, 2009) sino también multiobjetivo (Bouacha, *et al.*, 2010; Horng *et al.*, 2008; Aslan *et al.*, 2007).

1.4.2 Técnicas de búsqueda heurística.

Las técnicas de búsquedas heurísticas son herramientas que imitan el comportamiento humano o de la naturaleza para encontrar valores óptimos de determinados problemas. Se destacan por su capacidad para tratar con problemas complejos y con funciones discontinuas, no suaves y multimodales. Dentro de ellas se destacan el algoritmo de enjambre de partículas, el de hormiguero, el de recocido simulado y los algoritmos evolutivos.

1.4.2.1 Algoritmo de enjambre de partículas.

La optimización por enjambre de partícula (*particle swarm optimization*, PSO) es una técnica estocástica de optimización basada en el uso de una población. Se inspira en el comportamiento de las bandadas de aves, en el cual cada individuo tiene su propia posición y velocidad, en cada instante dado, aunque se dirigen a un lugar común.

Los principales pasos del algoritmo de enjambre de partículas pueden describirse de la siguiente forma:

1. Crear la población inicial ($k = 0$), asignando aleatoriamente las posiciones, x_i^k , y velocidades, v_i^k , de cada individuo. Evaluar la función objetivo para cada individuo y tomarla como las respectivas mejores posiciones, b_i , y la mejor posición global, B .
2. Determinar la nueva posición, x_i^{k+1} , y velocidad, v_i^{k+1} , para cada individuo, utilizando las expresiones:

$$v_i^{k+1} = wv_i^k + c_1 \text{rnd}_{[0,1]}(b_i - x_i^k) + c_2 \text{rnd}_{[0,1]}(B - x_i^k) \quad (1.12a)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (1.12b)$$

donde w es la función de ponderación, $\text{rnd}_{[0,1]}$ es un número aleatorio en el rango $[0, 1]$ y c_1 , y c_2 son las velocidades de aprendizaje.

3. Evaluar cada individuo y si las posiciones individuales y globales son mejores que las almacenadas, actualizar b_i y B .
4. Hacer $k = k + 1$. Si $k < k_{\text{MAX}}$ volver al paso 2.

Dentro de las aplicaciones de algoritmo de enjambre de partículas a la optimización del torneado se encuentran (Karpát y Özel, 2007; Cardi *et al.*, 2008; Özel, 2009).

1.4.2.2 Algoritmo de hormiguero.

El algoritmo de hormiguero (*ant colony algorithm*, ACO) es una técnica probabilística de solución problemas computacionales que puedan reducirse a encontrar buenas rutas a través de grafos. Se inspira en el comportamiento de las hormigas, las cuales a pesar de sus limitadas habilidades cognitivas, son capaces de, colectivamente, encontrar buenas rutas en el transporte de comida.

Se han reportado varias aplicaciones del algoritmo de hormiguero a la optimización de procesos de maquinado (Ghaiebi y Solimanpur 2007; Vijayakumar *et al.* 2003), pero también se han publicado críticas que cuestionan la eficiencia de este método (Wang, 2007).

1.4.2.3 Recocido simulado.

El recocido simulado (*simulated annealing*, SA) es una técnica global de optimización utilizada para lograr una buena aproximación al óptimo global de una función dada, en un espacio grande de búsqueda. Es, a menudo, usado cuando el espacio de búsqueda es discreto. El nombre y la inspiración provienen del endurecimiento por calor en la metalurgia, que mediante el calentamiento y el enfriamiento controlado de un material aumenta el tamaño sus cristales y reduce sus defectos.

Por la analogía con este proceso físico el recocido simulado atraviesa el espacio de búsqueda generando soluciones cercanas con de la solución actual. En cada paso se reemplaza la solución actual por una solución cercana aleatoria, seleccionada con una probabilidad que depende de la diferencia entre la función correspondiente y en un parámetro global t (llamado temperatura, por analogía), que es gradualmente disminuida durante el proceso. Una nueva solución superior es siempre aceptada. Una nueva solución inferior es aceptada probabilísticamente, basándose en la diferencia en la calidad y un parámetro de temperatura.

Dentro de las aplicaciones a optimización de procesos de corte, con recocido simulado se destacan (Saravanan *et al.*, 2003; Wang *et al.* 2004; Raja y Baskar, 2010).

1.4.2.4 Algoritmos evolutivos.

Los algoritmos evolutivos son un conjunto de heurísticas que simulan el proceso de evolución natural. Aunque el mecanismo en el que se basan es simple, estos algoritmos han probado ser una herramienta de búsqueda general, robusta y poderosa. En particular, son especialmente convenientes para problemas con múltiples objetivos conflictivos y espacios de búsqueda grandes y complejos.

A pesar de la gran variedad de enfoques propuestos, cualquier algoritmo evolutivo puede caracterizarse por tres rasgos fundamentales:

1. se mantiene un conjunto de soluciones candidatas;
2. se efectúa un proceso de selección por mezclado sobre este conjunto;
3. pueden combinarse varias soluciones, en términos de recombinación, para generar nuevas soluciones.

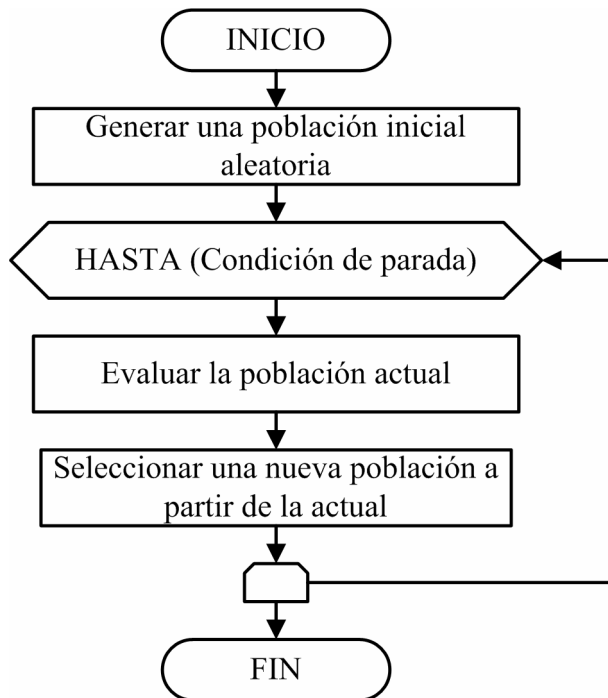


Figura 1.5 Diagrama de bloques de un algoritmo evolutivo típico.

Hay dos heurísticas principales dentro de los algoritmos evolutivos: la escuela alemana, con las estrategias evolutivas (*evolution strategies*, ES) y la escuela norteamericana, con los algoritmos genéticos (*genetic algorithms*, AG). En la Tabla 1.1 se resumen las principales diferencias entre estos enfoques.

Tabla 1.1 Diferencias entre ES y AG.

<i>Característica</i>	<i>Estrategias evolutivas (ES)</i>	<i>Algoritmos genéticos (GA)</i>
Codificación individual	Como número real	Como cadena de bits
Mutación	Adicionando un valor aleatorio	Cambiando un bit
Cruzamiento	Combinando números reales	Combinando trozos de las cadenas de bits

Los algoritmos evolutivos y, en especial, los algoritmos genéticos, han sido ampliamente usados en la optimización de procesos de corte de metales. Se pueden destacar, dentro de las más recientes (Chien y Tsai, 2003; Yildiz y Ozturk, 2006; Quiza *et al.*; 2006; Cus y Zuperl, 2006; Quiza *et al.*, 2009).

CAPÍTULO 2. DESARROLLO DE LA APLICACIÓN.

En este Capítulo se define la metodología de optimización de torneado propuesta, así como su implementación en forma de librería de funciones de MATLAB.

2.1 Definición de la metodología propuesta.

2.1.1 Variables de decisión.

Dentro del problema considerado, las variables de decisión son los componentes del régimen de corte. Para toda pasada de desbaste, son usados los mismos parámetros de corte, de ahí que son seis variables de decisión:

- Profundidad de corte para las pasadas de acabado, a_F .
- Número de pasadas, N .
- Avance para las pasadas de desbaste, f_R .
- Avance para las pasadas de acabado, f_F .
- Velocidad de corte para las pasadas de desbaste, v_R .
- Velocidad de corte para las pasadas de acabado, v_F .

A partir de los valores anteriores, pueden determinarse los parámetros de corte para cada pasada, según las expresiones:

$$a_i = \begin{cases} \frac{d_0 - d - 2a_F}{2(N-1)} & : i = 1, \dots, N-1 \\ a_F & : i = N \end{cases} ; \quad (2.1)$$

$$f_i = \begin{cases} f_R & : i = 1, \dots, N-1 \\ f_F & : i = N \end{cases} ; \quad (2.2)$$

$$v_i = \begin{cases} v_R & : i = 1, \dots, N-1 \\ v_F & : i = N \end{cases} . \quad (2.3)$$

2.1.2 Función objetivo.

El propósito de la optimización es minimizar simultáneamente dos objetivos contradictorios. El primer objetivo es el tiempo de maquinado por pieza, τ , el cual puede ser considerado como la suma de todos los tiempos auxiliares, τ_S , el tiempo total de corte, τ_C , el tiempo de cambio de la herramienta, τ_T , y el tiempo de movimiento de retorno de la herramienta, τ_B :

$$\tau = \tau_S + \tau_C + \tau_T + \tau_B ; \quad (2.4a)$$

La expresión anterior puede ser escrita como:

$$\tau = \tau_S + \sum_{i=1}^N \tau_{Ci} + \tau_{T0} \sum_{i=1}^N \frac{\tau_{Ci}}{T_i} + (N-1)\tau_{B0} ; \quad (2.4b)$$

donde τ_{Ci} es el tiempo de corte para la i -ésima pasada:

$$\tau_{Ci} = \frac{\pi d_{i-1} L}{1000 v_i f_i} ; \quad (2.5)$$

τ_{T0} es el tiempo de cambio de una herramienta, y τ_{B0} es el tiempo de movimiento de retorno de la herramienta, la vida útil de la herramienta para cada pasada, T_i , puede computarse usando la ley extendida de Taylor:

$$T_i = C_T v_i^{\alpha_1} f_i^{\alpha_2} a_i^{\alpha_3} . \quad (2.6)$$

El segundo objetivo es el gasto total de la herramienta, ξ , considerado como la parte de de la herramienta que se consume en el proceso:

$$\xi = \sum_{i=1}^N \xi_i = \sum_{i=1}^N \frac{\tau_i}{T_i} ; \quad (7)$$

2.1.3 Restricciones.

Para el proceso de corte, varias restricciones deben ser satisfechas. Primero que todo, los parámetros de corte, al ser seleccionados, deben estar en un rango permitido, en correspondencia con la herramienta de corte usada:

$$a_i^{MIN} \leq a_i \leq a_i^{MAX}; \quad \forall i \in [1, \dots, N]; \quad (8a)$$

$$f_i^{MIN} \leq f_i \leq f_i^{MAX}; \quad \forall i \in [1, \dots, N]; \quad (8b)$$

$$v_i^{MIN} \leq v_i \leq v_i^{MAX}; \quad \forall i \in [1, \dots, N]. \quad (8c)$$

Además, la fuerza de corte para cada pasada (F_i), no puede superar el valor máximo (F_{MAX}) permitido por la máquina herramienta:

$$F_i = C_F v_i^{\beta_1} f_i^{\beta_2} a_i^{\beta_3} \leq F_{MAX}; \quad \forall i \in [1, \dots, N]. \quad (2.9)$$

Otra restricción es que la potencia de corte P_i , no supere la que pueda ofrecer el motor de la máquina herramienta, P_{MOT} , teniendo en cuenta la eficiencia de la transmisión, η :

$$\frac{v_i F_i}{6 \times 10^4} \leq \eta P_{MOT}; \quad \forall i \in [1, \dots, N]. \quad (2.10)$$

Las dos restricciones anteriores deben analizarse especialmente para las pasadas de desbaste.

Finalmente, en la pasada de acabado, la rugosidad superficial obtenida (R_i) debe ser menor o igual al valor especificado (R_{MAX}), dado por el criterio tecnológico:

$$R_i = \frac{125f_i^2}{r_E} \leq R_{MAX} : i = N ; \quad (2.11)$$

donde r_E es el radio de la punta de la herramienta.

2.1.4 Algoritmo genético.

Para llevar a cabo el proceso de optimización ha sido usado un micro algoritmo genético (micro-GA), propuesto por Coello y Toscano (2001), con algunas modificaciones introducidas por Quiza *et al.*, (2006). Éste es un algoritmo elitista que usa una población adicional para mantener las soluciones más convenientes.

Otro rasgo importante del micro-GA es el uso de una población estática que se genera al principio del proceso y sigue sin modificación a través de la evolución. Esta población estática permite guardar una diversidad adecuada, a un costo computacional bajo.

El diagrama de bloques del algoritmo genético propuesto se muestra en la Fig. 2.1.

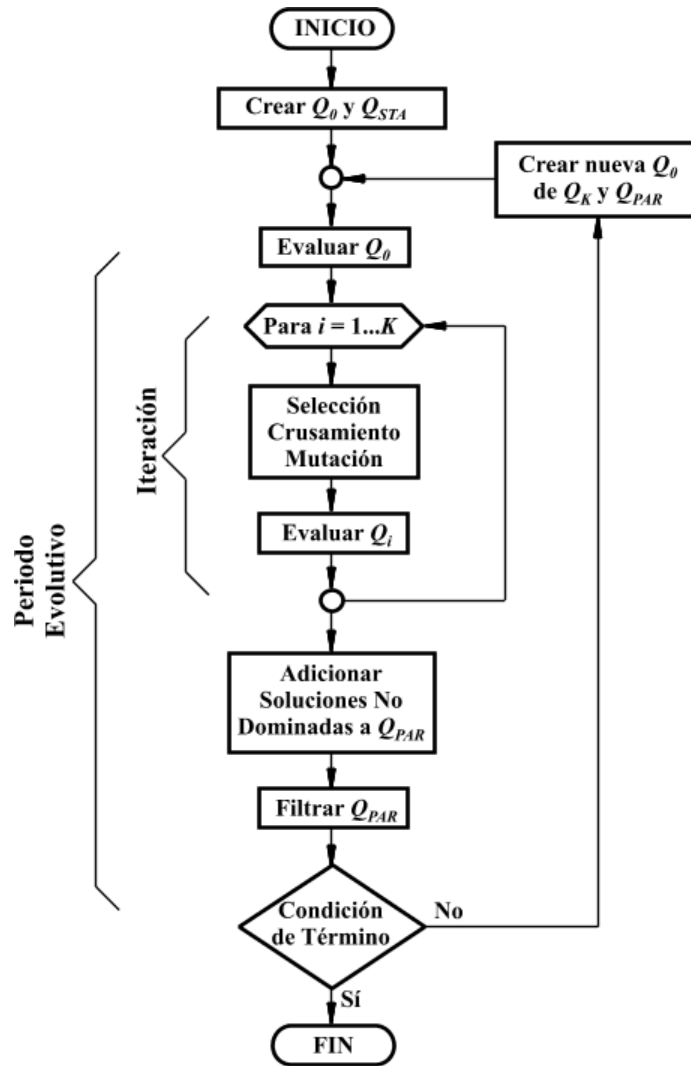


Figura 2.1 Diagrama de bloques del algoritmo genético utilizado.

El primer paso del micro-GA es la generación aleatoria de las dos poblaciones iniciales: la estática y dinámica. La población estática es varias veces mayor que la dinámica.

Los valores de todas las variables de decisión se codifican, para formar un cordón binario, llamado "cromosoma". La población dinámica inicial es evaluada, para calcular los valores correspondientes de ambos

objetivos de optimización: tiempo de maquinado y vida útil consumida de la herramienta.

A partir de esta población dinámica actual, es generada una nueva, aplicando los operadores de selección, cruzamiento y mutación. La selección se lleva a cabo por torneo, seleccionando dos parejas de progenitores. En cada pareja ambos candidatos son comparados, considerando el criterio de óptimo de Pareto. El cruzamiento combina el cordón de códigos de ambos "progenitores" formando un nuevo cordón para la nueva solución del "descendiente". En este enfoque, es implementado el cruzamiento de dos puntos. Finalmente, la mutación introduce algunos cambios aleatorios en el código de las nuevas soluciones. Aunque no hay garantía sobre la conveniencia de estos cambios, ellos ayudan a introducir diversidad en la población. Sin embargo, la probabilidad de la mutación debe ser bastante baja para evitar que el algoritmo genético degenera en una búsqueda aleatoria.

El proceso antedicho se considera una etapa. La nueva población se evalúa y se usa, en la próxima etapa, para obtener la siguiente. Después de alcanzado un cierto número de etapas, las soluciones no dominadas se extraen de la población resultante. Ellas se agregan a un grupo elitista de soluciones, llamada población paretiana que se filtra para mantener un cierto número predefinido de individuos, distribuidos tan uniformemente como sea posible.

El ciclo referido se llama período evolutivo. Después de eso, una nueva población dinámica se genera combinando las soluciones de la última población dinámica y de la estática. El período evolutivo se repite hasta lograr un frente de Pareto completo y uniforme.

2.2 Implementación de la metodología propuesta.

2.2.1 Función *creapop*.

La función *creapop* permite generar la población inicial, de forma aleatoria. Requiere como parámetros el tamaño de la población y, de cada una de las variables de decisión, el nombre, el valor mínimo, el máximo y la longitud del código. La variable de decisión es un arreglo de estructura de datos que contiene tres campos: valor mínimo (*MinVal*), valor máximo (*MaxVal*), longitud de codificación (*Length*).

El algoritmo de la función se muestra en la Fig. 2.2. Como se puede observar, se realiza un lazo doble, que recorre primeramente cada uno de los individuos de la población y, luego, cada una de las variables de decisión de cada individuo. Dentro del lazo se generan valores aleatorios en el rango válido para cada variable y se codifican para formar la cadena genotípica.

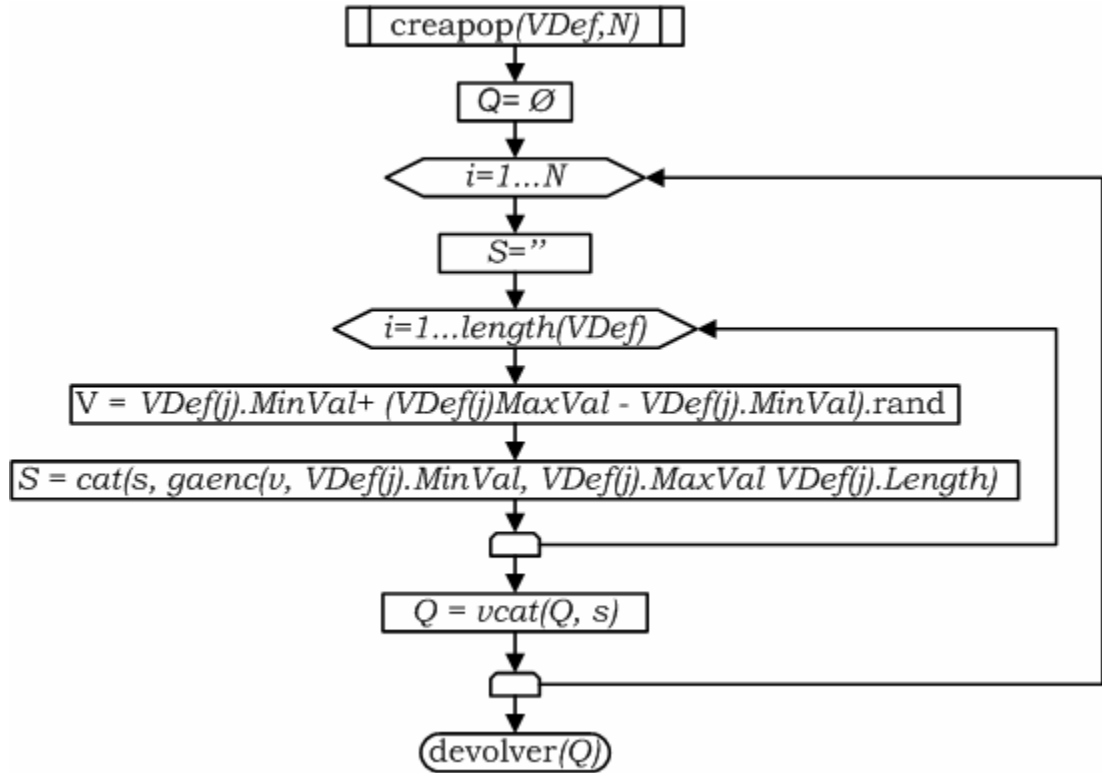


Figura 2.2 Diagrama de bloques de la función *creapop*.

El código correspondiente, se muestra en el List. 2.1. Esta función se utiliza para crear tanto la población dinámica inicial como estática.

Listado 2.1 Código de la función *creapop*.

```

function Q = creapop(VDef, N);
% Crea la población inicial

Q = '';
for I = 1 : N
    S = '';
    for J = 1 : length(VDef)
        V = (VDef(J).MaxVal - VDef(J).MinVal).*rand(1, 1) +
VDef(J).MinVal;
        S = strcatex(S, gaenc(V, VDef(J).MinVal,
VDef(J).MaxVal, VDef(J).Length));
    end
    Q = strvcat(Q, S);
end
    
```


2.2.2 Función *gaenc*.

La función *gaenc* permite codificar las variables de decisión en un valor de cadena genética. Requiere como parámetros el valor actual, el valor mínimo, el máximo y la longitud del código. Los tres primeros valores corresponden a números reales mientras que el último es un entero.

El algoritmo de la función se muestra en la Fig. 2.3. Como se puede observar, se realiza una secuencia simple, que primeramente convierte a un número entero en el rango de $[0.. 2^{CL}]$, por interpolación lineal, luego convierte el número anterior a cadena de texto de su representación binaria, y se crea un ciclo para completar con '0' hasta la longitud de codificación (CL).

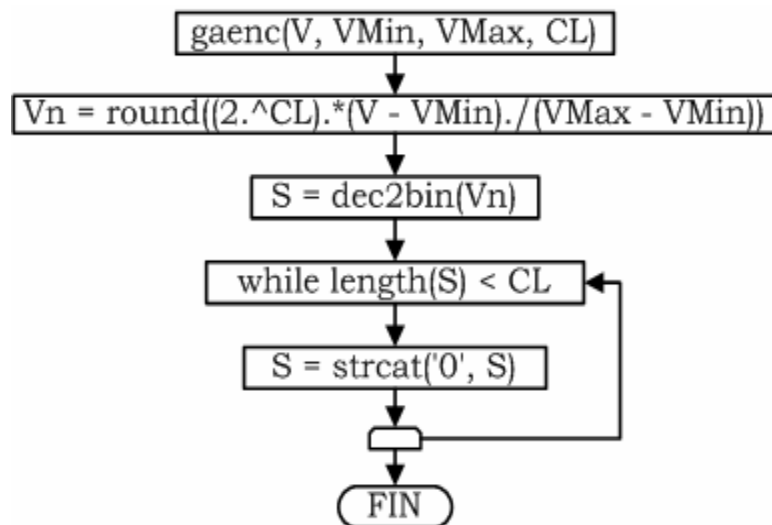


Figura 2.3 Diagrama de bloques de la función *gaenc*.

El código correspondiente, se muestra en el List. 2.2. Esta función se utiliza para codificar un valor en cadena genética.

Listado 2.2 Código de la función *gaenc*.

```
function S = gaenc(V, VMin, VMax, CL);
% Codifica un valor en cadena genética

Vn = round((2.^CL).*(V - VMin)./(VMax - VMin));
S = dec2bin(Vn);
while length(S) < CL
    S = strcat('0', S);
end
```

2.2.3 Función *gadec*.

La función *gadec* permite descodificar una cadena genética en sus valores, crea una función que devuelve una matriz o vector *V*. Requiere como parámetros el valor de la función codificado y la variable definida. La variable de decisión es un arreglo de estructura de datos que contiene cuatro campos: valor mínimo (*MinVal*), valor máximo (*MaxVal*), longitud de codificación (*Length*) y la variable definida (*VDef*).

El algoritmo de la función se muestra en la Fig. 2.4. Como se puede observar, se realiza un lazo simple, que recorre primeramente cada una de las variables de decisión definidas, después toma la porción de la cadena de codificación correspondiente a la variable, la convierte a valor numérico decimal y la lleva al intervalo $[V_{\min}, V_{\max}]$ por interpolación lineal.

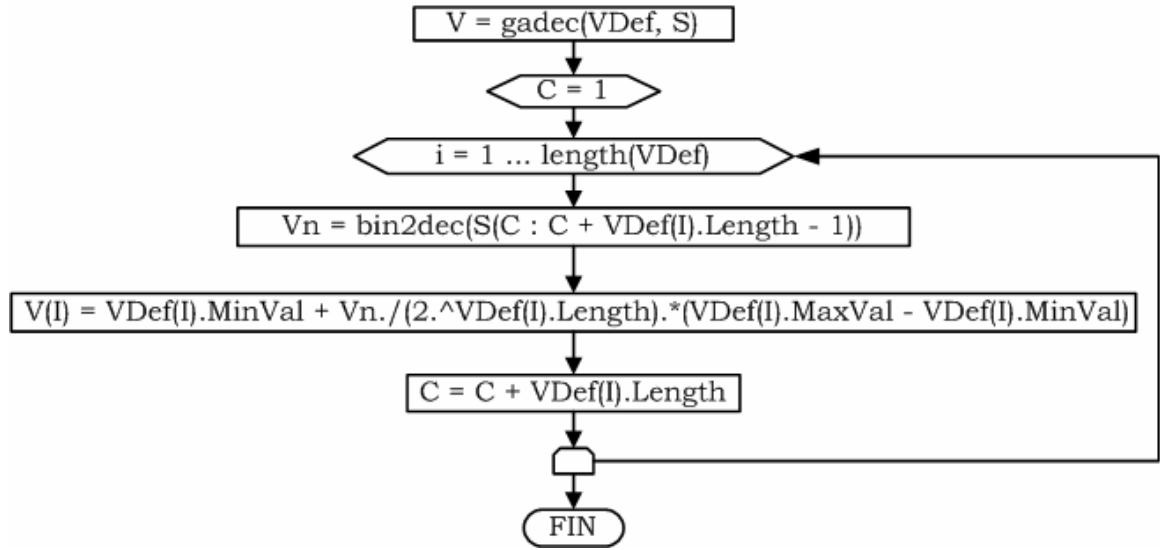


Figura 2.4 Diagrama de bloques de la función **gadec**.

El código correspondiente, se muestra en el List. 2.3. Esta función se utiliza para codificar un valor en cadena genética.

Listado 2.3 Código de la función *gadec*.

```

function V = gadec(VDef, S);
% Decodifica una cadena genética en sus valores

C = 1;
for I = 1 : length(VDef)
    Vn = bin2dec(S(C : C + VDef(I).Length - 1));
    V(I) = VDef(I).MinVal + Vn./(2.^VDef(I).Length).*(VDef(I).MaxVal - VDef(I).MinVal);
    C = C + VDef(I).Length;
end;
    
```

2.2.4 Función defpop.

La función *defpop* permite definir las variables de decisión. Requiere como parámetros el nombre, el valor mínimo, el máximo y la longitud del código. La variable de decisión que en el parámetro devuelto, es un

arreglo de estructura de datos que contiene tres campos: valor mínimo (*MinVal*), valor máximo (*MaxVal*), longitud de codificación (*Length*).

El algoritmo de la función se muestra en la Fig. 2.5. Como se puede observar, se realiza un lazo simple, que recorre cada uno de los valores asignados para las variables de decisión y asigna los valores a cada campo de la variable.

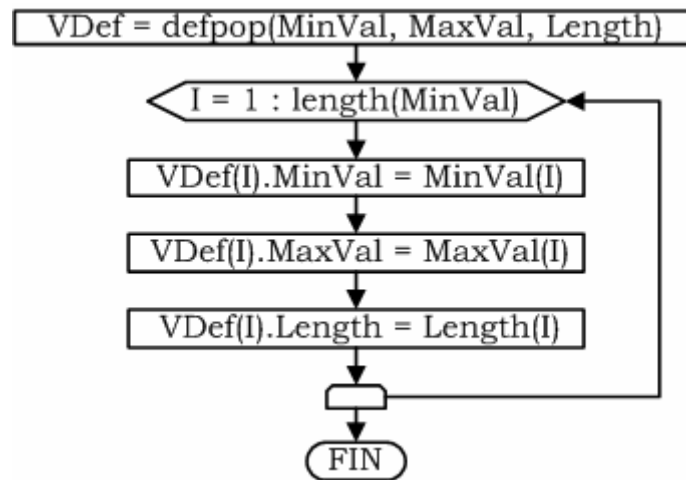


Figura 2.5 Diagrama de bloques de la función **defpop**.

El código correspondiente, se muestra en el List. 2.4.

Listado 2.4 Código de la función **defpop**.

```

function VDef = defpop(MinVal, MaxVal, Length);
% Define las variables de decisión

for I = 1 : length(MinVal)
    VDef(I).MinVal = MinVal(I);
    VDef(I).MaxVal = MaxVal(I);
    VDef(I).Length = Length(I);
end
  
```

2.2.5 Función evalpop.

La función *evalpop* permite evaluar la población, a partir de los valores de sus variables de decisión, obteniendo los valores de las fusiones objetivos y las restricciones. Requiere como parámetros el nombre, la variable definida, la población inicial y la función donde se va a evaluar. La variable de decisión (*Vdef*), es el arreglo antes mencionado; la población, (*Q*), es un arreglo de cadena de caracteres que contiene los cromosomas de los individuos; y la función, (*f*), es un puntero.

El algoritmo de la función se muestra en la Fig. 2.6. Como se puede observar, se realiza un lazo simple, que recorre cada uno de los individuos de la población y, luego, decodifica los valores de la variable de decisión del individuo. Luego evalúa las funciones objetivos para el individuo.

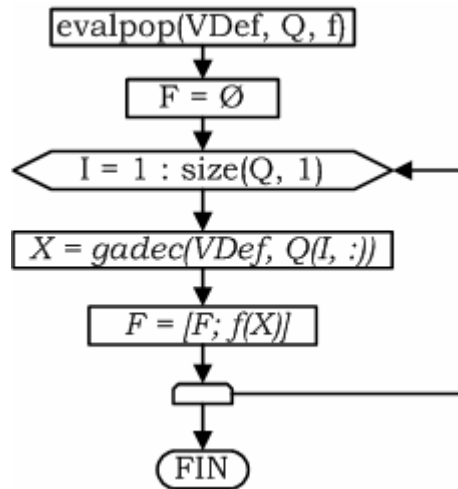


Figura 2.6 Diagrama de bloques de la función **evalpop**.

El código correspondiente, se muestra en el List. 2.5.

Listado 2.5 Código de la función evalpop.

```
function F = evalpop(VDef, Q, f);  
% Evalúa la población en una función  
  
F = [];  
for I = 1 : size(Q, 1)  
    X = gadec(VDef, Q(I, :));  
    F = [F; f(X)];  
end
```

2.2.6 Función *seltorn*.

La función *seltorn* permite seleccionar los nuevos progenitores por torneo. Requiere como parámetros el nombre, la población inicial, Q, el puntero a la función objetivo, F , y el puntero a las restricciones, G.

El algoritmo de la función se muestra en la Fig. 2.7. Como se puede observar, se realiza un lazo doble, primeramente inicializa con los progenitores como conjuntos vacíos, luego recorre todos los individuos de la población, selecciona N_1 y N_2 entre la cantidad de individuos aleatoriamente y se le aplica las condiciones de dominación entre los individuos, después se le agrega los individuos a la población de progenitores.

Listado 2.6 Código de la función `seltorn`.

```

function [P1, P2] = seltorn(Q, F, G);
% Selecciona los nuevos progenitores por torneo

P1 = [];
P2 = [];
for I = 1 : size(Q, 1)
    for J = 1 : 2
        N1 = round(1 + rand.*(size(Q, 1) - 1));
        N2 = N1;
        while N2 == N1
            N2 = round(1 + rand.*(size(Q, 1) - 1));
        end
        if isfeas(G(N1, :)) & ~isfeas(G(N2, :))
            % N1 is feasible and N2 is unfeasible
            P(J) = N1;
        elseif ~isfeas(G(N1, :)) & isfeas(G(N2, :))
            % N1 is unfeasible and N2 is feasible
            P(J) = N2;
        elseif isfeas(G(N1, :)) & isfeas(G(N2, :))
            % Both, N1 and N2, are feasible
            if dominate(F(N1, :), F(N2, :))
                % N1 dominates N2
                P(J) = N1;
            elseif dominate(F(N2, :), F(N1, :))
                % N2 dominates N1
                P(J) = N2;
            else
                % Both are non-dominated
                P(J) = N1;
            end
        end
    else
        % Neither N1 nor N2 are feasible
        if (sum(G(N1, :)) < sum(G(N2, :)))
            % N1 is less unfeasible than N2
            P(J) = N1;
        else
            % N2 is less unfeasible than N1
            P(J) = N2;
        end
    end
end
end
P1 = strvcatex(P1, Q(P(1), :));
P2 = strvcatex(P2, Q(P(2), :));
end

```


2.2.7 Función **crossover**.

La función *crossover* permite el cruzamiento de los nuevos valores con los ya conocidos del proceso de torneado. Requiere como parámetros, los progenitores, P_1 y P_2 . Devuelve la población generada, Q .

El algoritmo de la función se muestra en la Fig. 2.8. Como se puede observar, se realiza un lazo simple, que recorre todos los progenitores y selecciona aleatoriamente el punto de ruptura, luego combina ambos segmentos de cadena de código.

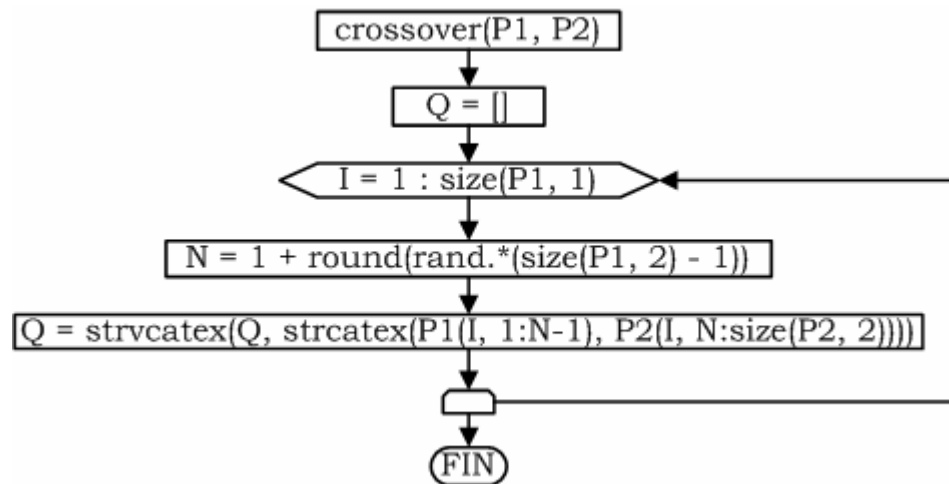


Figura 2.8 Diagrama de bloques de la función **crossover**

El código correspondiente, se muestra en el List. 2.7.

Listado 2.7 Código de la función *crossover*.

```

function Q = crossover(P1, P2);

Q = [];
for I = 1 : size(P1, 1)
    N = 1 + round(rand.*(size(P1, 2) - 1));
  
```

```
Q = strvcatex(Q, strcatex(P1(I, 1:N-1), P2(I, N:size(P2,  
2)))));  
end
```

2.2.8 Función *mutate*.

La función *mutate* permite aplicar la mutación a la población resultante entre la nueva generación. Requiere como parámetro la probabilidad de mutación, p , que por defecto es igual a 10^{-4} . La población inicial, Q . El algoritmo de la función se muestra en la Fig. 2.9. Como se puede observar, se realiza un lazo doble, que recorre primeramente cada uno de los individuos de la población y, luego, recorre la longitud de cadena y genera un número aleatorio y si es menor que la probabilidad de mutación cambia el carácter correspondiente de la cadena.

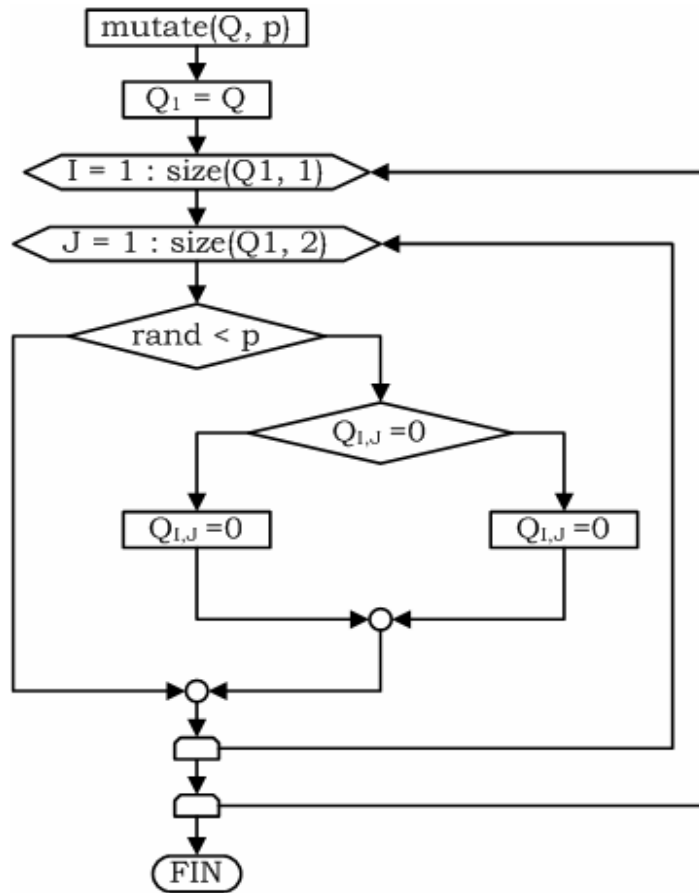


Figura 2.9 Diagrama de bloques de la función *mutate*

El código correspondiente, se muestra en el List. 2.8.

Listado 2.8 Código de la función *mutate*.

```

function Q1 = mutate(Q, p);

Q1 = Q;
for I = 1 : size(Q1, 1)
    for J = 1 : size(Q1, 2)
        if (rand < p)
            if strcmp(Q(I, J), '0')
                Q(I, J) = num2str('1');
            else
                Q(I, J) = num2str('0');
            end
        end
    end
end
end
end

```

2.2.9 Función *parpop*.

La función *parpop* permite extraer los elementos no dominados de la población anterior. Requiere como parámetros, la nueva población obtenida por el cruzamiento, *Q*, y los punteros a la función objetivo y las restricciones.

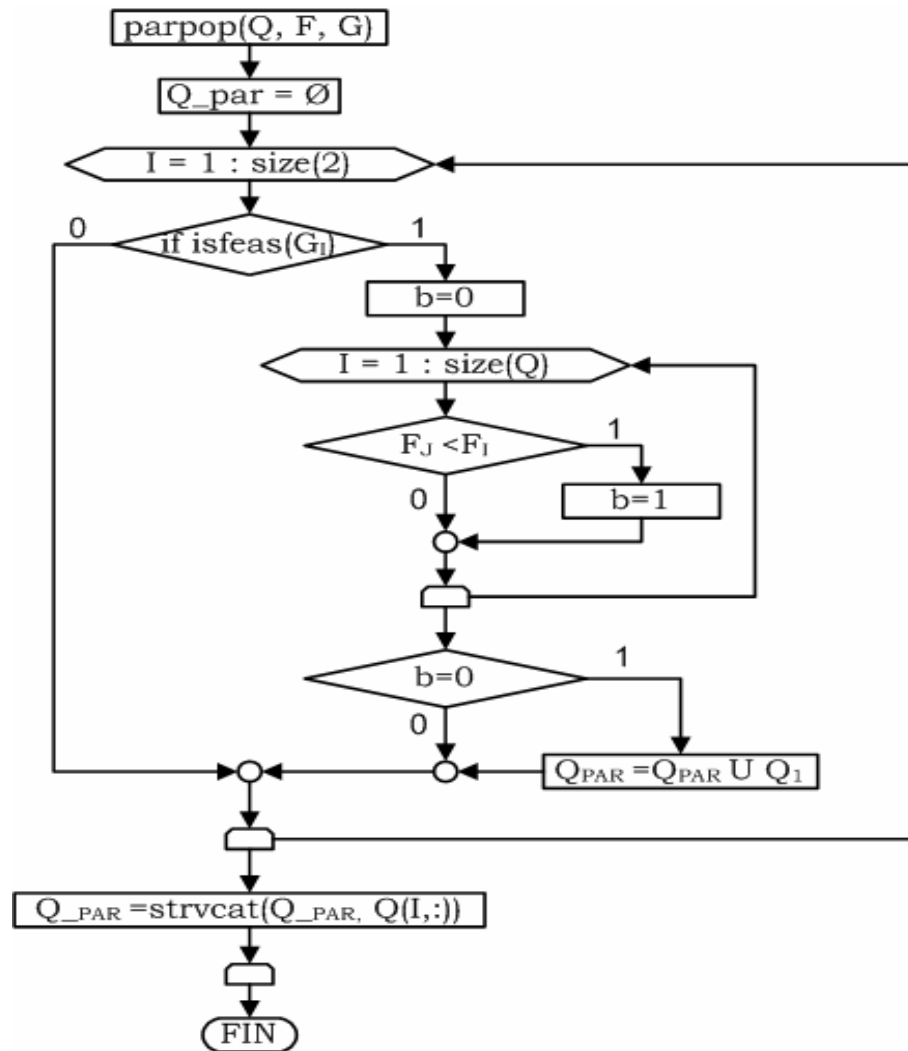


Figura 2.10 Diagrama de bloques de la función **parpop**.

El algoritmo de la función se muestra en la Fig. 2.10. Como se puede observar, se realiza un lazo doble, que recorre primeramente cada uno de los individuos de la población y, luego, compara cada uno de los elementos y los que son dominados por el otro individuo toman el valor de 1 y los no dominados toman el valor 0 y son incorporados a la población paretiana.

El código correspondiente, se muestra en el List. 2.9.

Listado 2.9 Código de la función *parpop*.

```
function Q_par = parpop(Q, F, G);
% Extrae los elementos no dominados de Q

Q_par = [];
for I = 1 : size(Q, 1)
    if isfeas(G(I, :))
        b = 0;
        for J = 1 : size(Q, 1)
            if dominate(F(J, :), F(I, :))
                b = 1;
            end
        end
        if (b == 0)
            Q_par = strvcat(Q_par, Q(I, :));
        end
    end
end
```

2.2.10 Función *filtpop*.

La función *filtpop* permite filtrar la población paretiana, dejando sólo un cierto número de individuos en ella. Requiere como parámetros la población paretiana, Q; el puntero a la función objetivo, F; y el tamaño máximo de la población paretiana, N. Devuelve la población filtrada, Q₁.

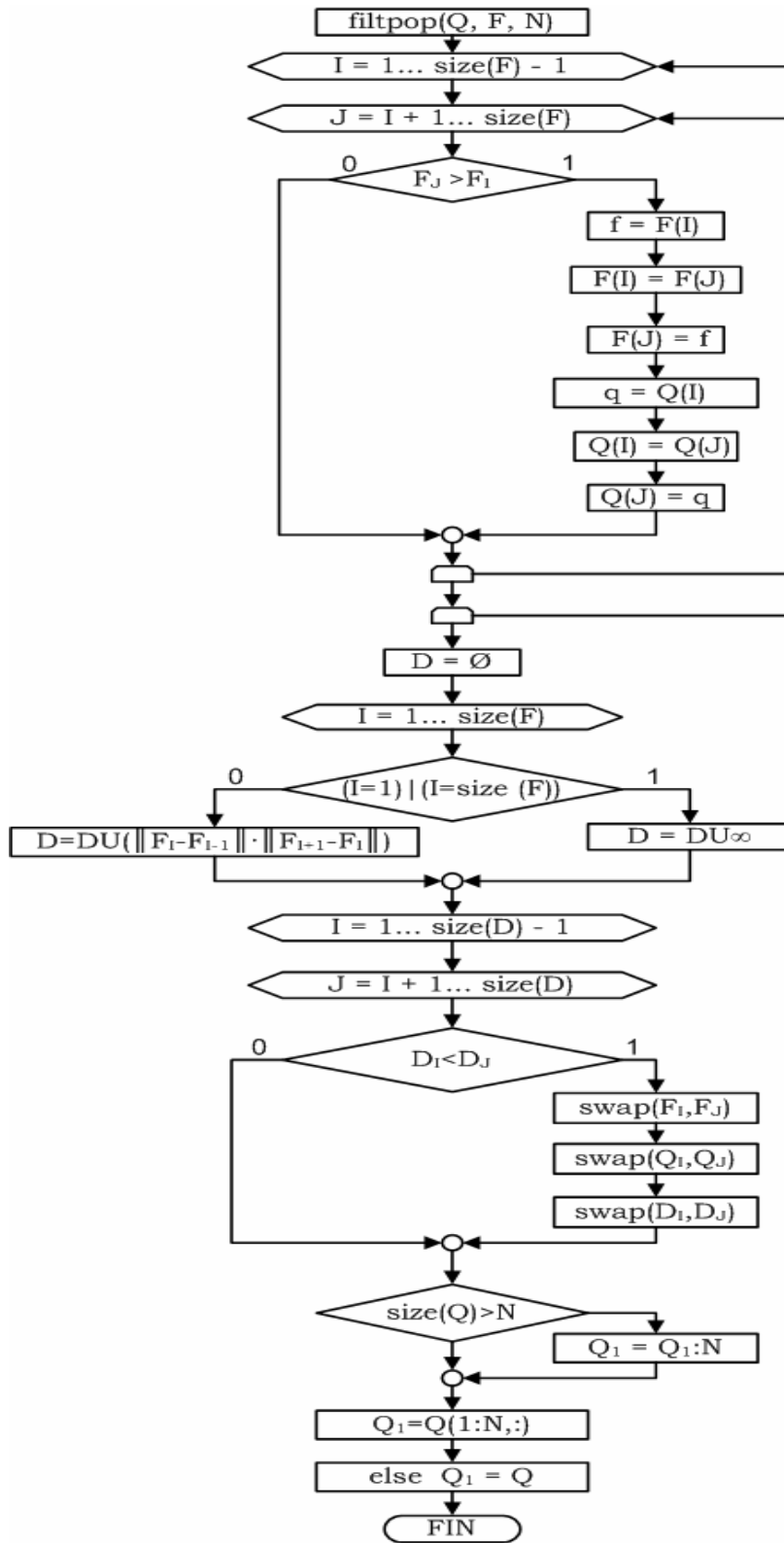


Figura 2.11 Diagrama de bloques de la función **filtpop**.

El algoritmo de la función se muestra en la Fig. 2.11. Como se puede observar, se realiza un lazo doble, que recorre primeramente cada uno de los individuos de la población y, luego, filtra los individuos de la población y extrae los elementos no dominados. El código correspondiente, se muestra en el List. 2.10.

Listado 2.10 Código de la función `filtpop`.

```
function Q1 = filtpop(Q, F, N);

for I = 1 : size(F, 1) - 1
    for J = I + 1 : size(F, 1)
        if F(I, 1) > F(J, 1)
            f = F(I, :);
            F(I, :) = F(J, :);
            F(J, :) = f;
            q = Q(I, :);
            Q(I, :) = Q(J, :);
            Q(J, :) = q;
        end
    end
end

D = [];
for I = 1 : size(F, 1)
    if (I == 1) | (I == size(F, 1))
        D = [D; Inf];
    else
        D = [D; sqrt(norm(F(I, :) - F(I - 1, :)).
*norm(F(I, :) - F(I + 1, :)))];
    end
end

for I = 1 : size(D, 1) - 1
    for J = I + 1 : size(D, 1)
        if D(I, 1) < D(J, 1)
            f = F(I, :);
            F(I, :) = F(J, :);
            F(J, :) = f;
            q = Q(I, :);
            Q(I, :) = Q(J, :);
            Q(J, :) = q;
        end
    end
end
```

```
        d = D(I, :);
        D(I, :) = D(J, :);
        D(J, :) = d;
    end
end
end

if size(Q, 1) > N
    Q1 = Q(1:N, :);
else
    Q1 = Q;
end
```


CAPÍTULO 3. ESTUDIOS DE CASO.

Para ilustrar el uso de las funciones desarrolladas, así como para verificar su correcta eficiencia y eficacia, se implementaron dos ejemplos de procesos de torneado, como estudios de casos.

3.1 Caso de estudio 1: Torneado de una sola pasada.

3.1.1 Definición del caso de estudio.

El primer caso de estudio corresponde al torneado de acabado de una aleación de titanio Ti-6Al-4V, con una dureza de 600 HV. El proceso será llevado a cabo en un torno de CNC, de potencia 20 kW, utilizando una herramienta de carburos ISO CNMG 12 04 08-883.

Para la vida útil de la herramienta, y para la rugosidad superficial, se han obtenido los siguientes modelos empíricos (Hasçalik y Çaydaş, 2007):

$$T = 1602 \frac{a^{0,3953}}{v^{1,322} f^{0,8686}} \quad (3.1)$$

$$R = 19,24v^{0,5151} f^{1,229} a^{0,6176} \quad (3.2)$$

La herramienta de corte seleccionada tiene los siguientes rangos de trabajo:

$$a_{\text{MIN}} = 0,5 \text{ mm}; \quad a_{\text{MAX}} = 2,0 \text{ mm}; \quad (3.3a)$$

$$f_{\text{MIN}} = 0,15 \text{ mm/rev}; \quad f_{\text{MAX}} = 0,35 \text{ mm/rev}; \quad (3.3b)$$

$$v_{\text{MIN}} = 30 \text{ m/min}; \quad v_{\text{MAX}} = 90 \text{ m/min}. \quad (3.3c)$$

En la operación considerada, un volumen, V , de 219 912 mm³ de material será removido. El tiempo preparativo-conclusivo, τ_s ; el tiempo de cambio de la herramienta, τ_c ; y el tiempo de trabajo en vacío, τ_0 ; se tomarán iguales a 0,15 min, 0,20 min y 0,05 min, respectivamente.

3.1.2 Implementación del problema.

Para la solución del problema se utilizaron las expresiones que se explican a continuación.

El tiempo total de elaboración, se determina por:

$$\tau = \tau_s + \frac{V}{M} \left(1 + \frac{\tau_c}{T} \right) + \tau_0; \quad (3.4)$$

donde M es la velocidad de remoción de material la cual, a su vez, se determina como:

$$M = 1000 \cdot v \cdot f \cdot a. \quad (3.5)$$

El gasto de herramienta, se calcula según:

$$\xi = \frac{V}{M \cdot T} \cdot 100\%; \quad (3.6)$$

mientras que la potencia de corte se obtiene de:

$$P_c = \frac{v \cdot F_c}{6 \cdot 10^4} \leq \frac{P_{MOT} \eta}{100}. \quad (3.8)$$

Para la solución del problema, se implementaron, primeramente, las funciones $f1$ y $g1$, que permiten calcular las funciones objetivos y las restricciones, respectivamente. Los códigos de estas funciones se muestran en los List. 3.1 y 3.2.

Listado 3.1 Código de la función de objetivos $f1$.

```
function y = f1(v, f, a);

V = 219912;
t_S = 0.15;
t_TC = 0.20;
t_0 = 0.05;

T = 1602.*a.^0.3953./(f.^0.8686.*v.^1.522);

M = 1000.*v.*f.*a;
t = t_S + V./M.*(1 + t_TC./T) + t_0;
z = 100.*V./(M.*T);
y = [t, z];
```

Listado 3.2 Código de la función de restricciones $g1$.

```
function g = g1(v, f, a);

Rmax = 12.5;

R = 19.24.*v.^0.5151.*f.^1.229.*a.^0.6176;
g = [R./Rmax - 1];
```

Por su parte, para la solución general del problema se seleccionó un tamaño de población de 250 individuos, las variables fueron codificadas en tramos de 48 bits, cada una, y se utilizó una probabilidad de mutación de $5 \cdot 10^{-2}$. El proceso fue detenido al cabo de 100 iteraciones.

La programación de la solución se muestra en el List. 3.3.

Listado 3.3 Código de solución del caso de estudio 1.

```
clear all;
InitialTime = rem(now, 1);
rand('state', sum(100*clock));
VDef = defpop([30 0.15 0.5], [90 0.35 2.0], [48 48 48]);
Q_eli = [];
f = inline('f3(x(1), x(2), x(3))');
g = inline('g3(x(1), x(2), x(3))');
```

```

Q = [];
while size(Q, 1) < 250
    q = creapop(VDef, 1);
    F = evalpop(VDef, q, f);
    G = evalpop(VDef, q, g);
    if isfeas(G)
        Q = [Q; q];
    end
end

F = evalpop(VDef, Q, f);
G = evalpop(VDef, Q, g);
for I = 1 : 100
    [P1, P2] = selturn(Q, F, G);
    Q = crossover(P1, P2);
    Q = mutate(Q, 0.005);
    F = evalpop(VDef, Q, f);
    G = evalpop(VDef, Q, g);
    Q_par = parpop(Q, F, G);
    Q_eli = strvcatex(Q_eli, Q_par);
    F_eli = evalpop(VDef, Q_eli, f);
    G_eli = evalpop(VDef, Q_eli, g);
    Q_eli = parpop(Q_eli, F_eli, G_eli);
    F_eli = evalpop(VDef, Q_eli, f);
    G_eli = evalpop(VDef, Q_eli, g);
    Q_eli = filtpop(Q_eli, F_eli, 25);
    fprintf('.');
    if (mod(I, 10) == 0), fprintf('\n');, end
end

F_eli = evalpop(VDef, Q_eli, f);
figure;
plot(F_eli(:,1), F_eli(:,2), '*r');

F = evalpop(VDef, Q_eli, f);
G = evalpop(VDef, Q_eli, g);
for I = 1 : size(Q_eli, 1)
    x = gadec(VDef, Q_eli(I, :));
    y = F(I, :);
    ga = G(I, :);
    g_(1) = (ga(1) + 1).*12.5;
    fprintf('%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',      x(1),
x(2), x(3), y(1), y(2), g_(1));
end

TotalTime = 24*60*60*(rem(now, 1) - InitialTime)

```

3.1.3 Análisis de los resultados.

Para la solución del problema se utilizó una computadora con microprocesador Mobile DualCore Intel Celeron a 1600 MHz, y 2 GiB de RAM. El código fue ejecutado utilizando MATLAB R2006a (7.2.0.232).

El proceso de evolución tomó un tiempo total de 183 s . Como resultado, se obtuvieron las soluciones que se muestran en la Tabla 3.1.

Tabla 3.1 Soluciones del caso de estudio 1.

No.	v [m/min]	f [mm/rev]	a [mm]	τ [min]	ξ [%]	R [μ m]
1	30.0	0.162	1.05	43.3	96.1	12.2
2	30.1	0.162	1.05	43.2	96.2	12.3
3	30.2	0.162	1.05	43.1	96.3	12.3
4	30.3	0.162	1.05	43.0	96.5	12.3
5	30.5	0.162	1.05	42.7	96.9	12.3
6	30.6	0.162	1.05	42.5	97.0	12.4
7	30.8	0.162	1.05	42.3	97.3	12.4
8	30.8	0.162	1.05	42.2	97.4	12.4
9	30.9	0.162	1.05	42.1	97.6	12.4
10	31.0	0.162	1.05	42.0	97.7	12.4
11	31.1	0.162	1.05	41.9	97.8	12.5
12	31.2	0.162	1.05	41.8	97.9	12.5
13	31.2	0.162	1.05	41.7	98.1	12.5
14	31.3	0.162	1.05	41.6	98.2	12.5
15	31.7	0.161	1.05	41.4	98.9	12.5
16	32.1	0.161	1.05	41.1	100.0	12.5
17	35.3	0.155	1.03	39.3	107.9	12.5
18	39.0	0.151	1.00	37.9	119.1	12.4

Como se puede observar (Fig. 3.1), se obtuvo una frontera de Pareto convexa, con una distribución aceptablemente regular.

Como se puede ver, los principales cambios se produjeron en la velocidad de corte, que varió en el rango de (30 ... 39) m /s. En cambio tanto el avance como la profundidad de corte, se mantuvieron prácticamente estables en todas las soluciones.

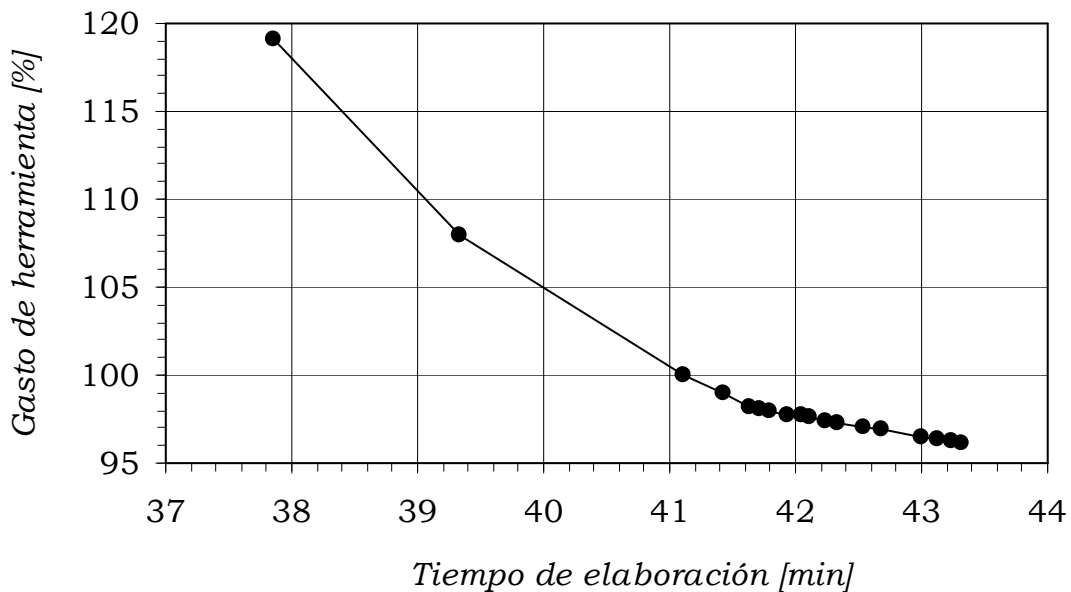


Figura 3.1 Frontera de Pareto para el caso de estudio 1.

La selección del régimen más conveniente depende de las necesidades reales de la unidad de producción. En caso de requerirse una alta productividad, se seleccionará el régimen correspondiente al punto 18, que es el que menor tiempo de elaboración presenta, aunque se destaca por el gasto herramental particularmente alto. Por el contrario, si las herramientas son deficitarias o excesivamente caras, se

preferirá el régimen de corte del punto 1, a pesar de su baja productividad. En otras situaciones, se seleccionará un punto intermedio, probablemente, con la aplicación de algún criterio económico concreto.

3.2 Caso de estudio 2: Torneado de pasadas múltiples.

3.2.1 Definición del caso de estudio.

Se desea realizar el torneado de un escalón de longitud $L = 180$ mm, en un árbol de acero AISI 1045, desde un diámetro $D_0 = 150$ mm hasta $D = 100$ mm. De acuerdo con los requisitos tecnológicos la superficie a obtener tendrá una rugosidad no mayor de $2,0 \mu\text{m}$, en Ra.

El proceso será llevado a cabo en un torno de CNC, con un motor de potencia $P_{\text{MOT}} = 10$ kW, y eficiencia de la transmisión, $\eta = 75\%$. Una fuerza máxima $F_{\text{MAX}} = 5\,000$ N, es permitida por el sistema máquina-dispositivo-herramienta-pieza. Tanto las pasadas de desbaste como las de acabado serán llevadas a cabo con una herramienta de cambio rápido con placa de carburo revestida ISO TNMG 22 04 16/P20, con un radio de la punta $r_E = 2,0$ mm.

Para la combinación herramienta-pieza utilizadas, se tienen las siguientes ecuaciones empíricas para la vida útil (Quiza *et al.*, 2009)

$$T_i = \frac{5.48 \times 10^9}{v_i^{3.46} f_i^{0.696} a_i^{0.460}}; \quad (3.9)$$

$$F_i = \frac{6.56 \times 10^3 f_i^{0.917} a_i^{1.10}}{v_i^{0.286}}. \quad (3.10)$$

Las ecuaciones anteriores sólo son válidas para los rangos experimentales siguientes:

$$0.5 \text{ mm} \leq a_i \leq 6.0 \text{ mm}; \quad (3.11a)$$

$$0.15 \text{ mm/rev} \leq f_i \leq 0.55 \text{ mm/rev}; \quad (3.11b)$$

$$250 \text{ m/min} \leq v_i \leq 400 \text{ m/min}. \quad (3.11c)$$

Adicionalmente, para la pasada de acabado, la profundidad de corte no debe superar el valor de 0,7, para garantizar una baja rugosidad superficial, o sea:

$$a_i \leq 0.7 \text{ mm}: i = N. \quad (3.11d)$$

El tiempo preparativo-conclusivo, τ_s ; el tiempo de cambio de la herramienta, τ_c ; y el tiempo de trabajo en vacío, τ_0 ; se asumen como 1,5 min; 1,0 min y 0,1 min, respectivamente.

3.2.2 Implementación del problema.

Para la solución del problema, se utilizarán las expresiones dadas en las secciones 2.1.2 y 2.2.3, del Capítulo 2.

Con el objetivo de calcular los valores de las funciones objetivos, así como de las restricciones del problema, se implementaron las funciones f_2 y g_2 , cuyo código se ofrece en los List. 3.4 y 3.5.

Listado 3.4 Código de la función de objetivos $f2$.

```

function y = f2(a_F, N, f_R, f_F, v_R, v_F);

D0 = 150;
D = 100;
L = 180;
c0 = 5.48e9;
c1 = -3.46;
c2 = -0.696;
c3 = -0.460;
V = 219912;
t_S = 1.5;
t_T0 = 1.0;
t_B0 = 0.1;

a(1:N-1) = (D0 - D - 2.*a_F)./(2.*(N - 1));
a(N) = a_F;
f(1:N-1) = f_R;
f(N) = f_F;
v(1:N-1) = v_R;
v(N) = v_F;

% T = c0.*v.^c1.*f.^c2.*a.^c3;
% M = 1000.*v.*f.*a;
t = t_S + (N - 1).*t_B0;
z = 0;
Di = D0;
for I = 1 : N
    Ti = c0.*v(I).^c1.*f(I).^c2.*a(I).^c3;
    t_Ci = pi.*Di.*L./(1000.*v(I).*f(I));
    t = t + t_Ci + t_T0.*t_Ci./Ti;
    Di = Di - 2.*a(I);
    z = z + t_Ci./Ti;
end
y = [t, 100.*z];

```

Listado 3.5 Código de la función de restricciones $g2$.

```

function g = g1(a_F, N, f_R, f_F, v_R, v_F);

D0 = 150;
D = 100;
L = 180;
d0 = 6.56e3;
d1 = -0.917;

```

```

d2 = 1.10;
d3 = 0.286;
Pm = 10;
n = 0.75;
Fmax = 5000;
Rmax = 2;
r_E = 2;

a(1:N-1) = (D0 - D - 2.*a_F)./(2.*(N - 1));
a(N) = a_F;
f(1:N-1) = f_R;
f(N) = f_F;
v(1:N-1) = v_R;
v(N) = v_F;

Fc = 0;
Pc = 0;
for I = 1 : N
    Fc = max(Fc, d0.*v(I).^d1.*f(I).^d2.*a(I).^d3);
    Pc = max(Pc, v(I).*Fc./6e4);
end
R = 125.*f(N).^2./r_E;
g = [100.*Pc./(n.*Pm) - 1, Fc./Fmax - 1, ...
     R./Rmax - 1, a(1)./6.0 - 1, 1 - a(1)./0.5];

```

Para la solución del problema, y teniendo en cuenta la mayor complejidad de la optimización de pasadas múltiples, se tomó una población de 500 individuos y se trabajó durante 150 iteraciones. El resto de los parámetros se mantuvo igual que en el caso de estudio anterior. El código de solución del problema, se muestra en el List. 3.6.

Listado 3.6 Código de solución del caso de estudio 2.

```

clear all;
InitialTime = rem(now, 1);
rand('state', sum(100*clock));
VDef = defpop([0.2  2    0.15 0.15 250  250], ...
              [0.7  20   0.55 0.55 400  400], ...
              [48   48   48   48   48   48], ...
              [1    0    1    1    1    1]);

```

```

Q_eli = [];
f = inline('f2(x(1), x(2), x(3), x(4), x(5), x(6))');
g = inline('g2(x(1), x(2), x(3), x(4), x(5), x(6))');

Q = [];
while size(Q, 1) < 500
    q = creapop(VDef, 1);
    F = evalpop(VDef, q, f);
    G = evalpop(VDef, q, g);
    if isfeas(G)
        Q = [Q; q];
    end
end
F = evalpop(VDef, Q, f);
G = evalpop(VDef, Q, g);

for I = 1 : 150
    [P1, P2] = seturn(Q, F, G);
    Q = crossover(P1, P2);
    Q = mutate(Q, 1e-4);
    F = evalpop(VDef, Q, f);
    G = evalpop(VDef, Q, g);
    Q_par = parpop(Q, F, G);
    Q_eli = strcatex(Q_eli, Q_par);
    F_eli = evalpop(VDef, Q_eli, f);
    G_eli = evalpop(VDef, Q_eli, g);
    Q_eli = parpop(Q_eli, F_eli, G_eli);
    F_eli = evalpop(VDef, Q_eli, f);
    G_eli = evalpop(VDef, Q_eli, g);
    Q_eli = filtpop(Q_eli, F_eli, 25);
    fprintf('.');
    if (mod(I, 10) == 0), fprintf('\n');, end
end

F_eli = evalpop(VDef, Q_eli, f);
figure;
plot(F_eli(:,1), F_eli(:,2), '*r');

F = evalpop(VDef, Q_eli, f);
G = evalpop(VDef, Q_eli, g);
for I = 1 : size(Q_eli, 1)
    x = gadec(VDef, Q_eli(I, :));
    y = F(I, :);
    ga = G(I, :);
    g_(1) = (ga(1) + 1).*7.5;
    g_(2) = (ga(2) + 1).*5000;
    g_(3) = (ga(3) + 1).*2;

```

```
fprintf('% .3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n', x(1), x(2), x(3), x(4), x(5), x(6), y(1), y(2), g_(1), g_(2), g_(3));
end

TotalTime = 24*60*60*(rem(now, 1) - InitialTime)
```

3.2.3 Análisis de los resultados.

Al igual que en el caso de estudio anterior, para la solución del problema se utilizó una computadora con microprocesador Mobile DualCore Intel Celeron a 1600 MHz, y 2 GiB de RAM. El código fue ejecutado utilizando MATLAB R2006a (7.2.0.232).

Tabla 3.2 Soluciones del caso de estudio 2.

No.	a_F [mm]	N	f_R [mm/rev]	f_F [mm/rev]	v_R [m/min]	v_F [m/min]	τ [min]	ξ [%]	P_c [kW]	F_c [W]	R [μ m]
1	0.20	11	0.45	0.18	254	254	10.24	21.63	7.5	1769	2.0
2	0.20	11	0.45	0.18	254	252	10.25	21.62	7.5	1769	2.0
3	0.20	11	0.45	0.18	254	251	10.26	21.61	7.5	1769	2.0
4	0.20	11	0.45	0.18	253	253	10.27	21.39	7.5	1772	2.0
5	0.20	11	0.45	0.18	253	252	10.28	21.38	7.5	1772	2.0
6	0.20	11	0.45	0.18	253	251	10.29	21.37	7.5	1772	2.0
7	0.20	11	0.45	0.18	252	253	10.30	21.16	7.5	1774	2.0
8	0.20	11	0.45	0.18	252	252	10.31	21.14	7.4	1774	2.0
9	0.20	11	0.45	0.18	250	253	10.32	20.95	7.5	1776	2.0
10	0.20	11	0.45	0.18	250	252	10.33	20.92	7.5	1776	2.0
11	0.20	11	0.45	0.18	250	251	10.34	20.90	7.4	1776	2.0
12	0.20	9	0.32	0.18	250	252	11.07	20.79	6.9	1639	2.0
13	0.20	9	0.32	0.18	250	251	11.08	20.78	6.8	1639	2.0

El proceso de evolución tomó un tiempo total de 993 s. Como resultado, se obtuvieron las soluciones que se muestran en la Tabla 3.2.

En la frontera de Pareto obtenida (ver Fig. 3.2), se puede observar las diferentes soluciones, que resultarán más convenientes en dependencia de las condiciones concretas de cada taller.

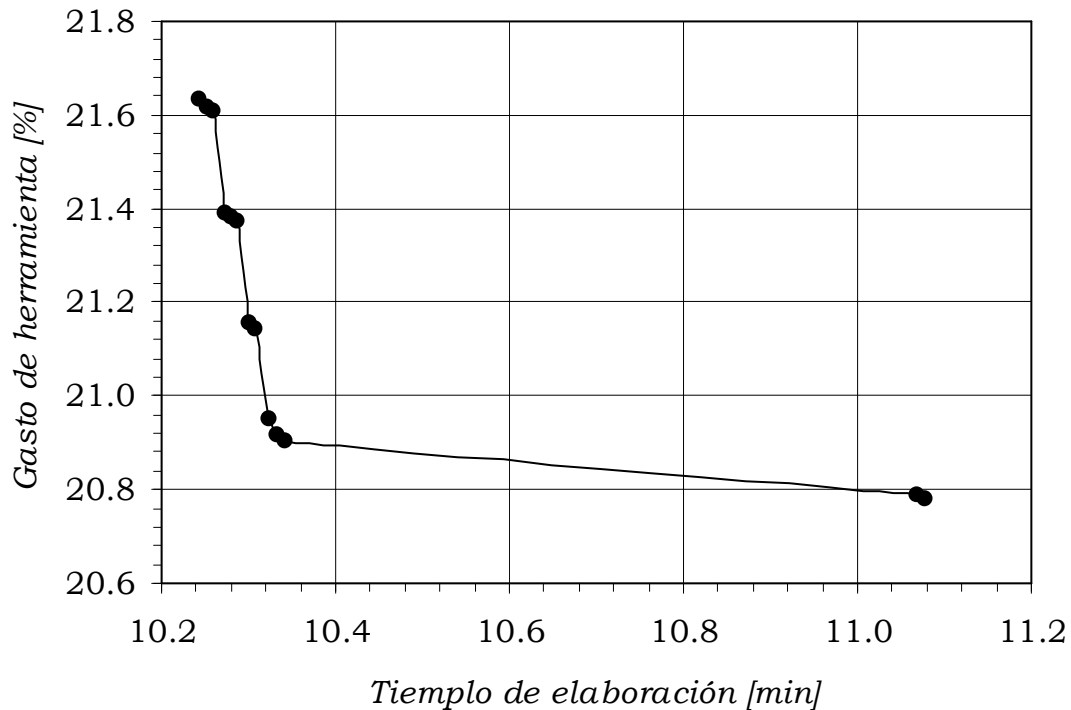


Figura 3.2 Frontera de Pareto para el caso de estudio 2.

De forma similar al caso anterior, si se desea un régimen con alta productividad, debe escogerse el punto 13. Por el contrario, el punto 1 garantiza el máximo aprovechamiento de la herramienta. Finalmente, los puntos intermedios son casos de compromiso entre los dos objetivos, que pudieran resultar más o menos convenientes, en dependencia de los costos de la operación.

3.3. Conclusiones parciales del capítulo.

A través de los estudios de casos presentados en el capítulo, se ha podido arribar a las siguientes conclusiones:

1. La librería de funciones desarrollada ha permitido obtener las soluciones de la optimización multiobjetivo, de regímenes de corte para torneado, tanto de una sola pasadas, como de pasadas múltiples, desde un enfoque a posteriori.
2. El funcionamiento de la biblioteca es estable y fiable, sin que se hayan detectado errores en su utilización.
3. El costo computacional del uso de la librería, en los casos estudiados, es aceptable, dada la complejidad de los problemas tratados.

CONCLUSIONES.

Como resultado de lo expuesto en el presente trabajo, se enuncian las siguientes conclusiones:

1. Se implementó una librería de funciones de MATLAB para resolver problemas de optimización multiobjetivo, de regímenes de corte para torneado, tanto de una sola pasadas, como de pasadas múltiples, desde un enfoque a posteriori.
2. La librería desarrollada, implementó un algoritmo genético multiobjetivo, con tratamiento de restricciones, que puede ser usado no sólo para el problema tratado, sino para cualquier problema de propósito general.
3. La librería implementada, según mostró el estudio de casos, tiene un funcionamiento estable, eficaz y razonablemente eficiente desde el punto de vista computacional.

RECOMENDACIONES.

Como resultado del trabajo se recomienda lo siguiente:

1. Incrementar la librería con otros algoritmos genéticos multiobjetivos y comparar su eficiencia con el actual.
2. Implementar el método de optimización propuesto para otros procesos de maquinado (fresado, taladrado, etc.).

BIBLIOGRAFÍA.

- Ahmad, N.; Anwarul Haque, A.F.M. (2001). "Optimization of process planning parameters for rotational components by genetic algorithms" [online], *4th Int. Conference on Mechanical Engineering*, Dhaka (Bangladesh), Vol. VII, pp. 227 – 233, [citado: 6-nov-2003], disponible en Internet en: <<http://www.buet.ac.bd/ipe/nafis/cv.pdf>>.
- Alfonso, E. (2002). "Apuntes sobre la interrelación composición-microestructura-propiedades del acero". *Monografías 2002* [cd-rom]. Matanzas: Universidad de Matanzas. ISBN 959-16-0188-3.
- Amiolemhen, E.; Ibhádode, A.O.A. (2004). "Application of genetic algorithms –determination of the optimal machining parameters in the conversion of a cylindrical bar stock into a continuous finished profile". *International Journal of Machine Tools and Manufacture* 44 (12 - 13), pp. 1403 - 1412.
- António, C.A.C.; Davim, J.P. (2002). "Optimal cutting conditions in turning of particulate metal matrix composites based on experiment and a genetic search model". *Composites Part A: Applied Science and Manufacturing* 33 (2), pp. 213 - 219.
- Aslan, E.; Camuscu, N.; Birgören, B. (2007). "Design optimization of cutting parameters when turning hardened AISI 4140 steel (63

- HRC) with Al₂O₃ + TiCN mixed ceramic tool”. *Materials and Design* 28(5), pp. 1618-1622.
- Basak, S.; Dixit, U.S.; Davim, J.P. (2007). “Application of radial basis function neural networks in optimization of hard turning of AISI D2 cold-worked tool steel with a ceramic tool”. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 221(6), pp. 987-998.
- Bhaskara Reddy, S.V.; Shunmugam, M.S.; Narendran, T.T. (1998). “Optimal subdivision of the depth of cut to achieve minimum production cost in multi-pass turning using a genetic algorithm”, *Journal of Materials Processing Technology* 79 (1-3), pp. 101 - 108.
- Bouacha, K.; Yallese, M.A.; Mabrouki, T.; Rigal, J.-F., (2010). “Statistical analysis of surface roughness and cutting forces using response surface methodology in hard turning of AISI 52100 bearing steel with CBN tool”. *International Journal of Refractory Metals and Hard Materials* 28(3), pp. 349-361.
- Cakir, M.C.; Gurarda, A. (1998). "Optimization and graphical representation of machining conditions in multi-pass turning operations". *Computer Integrated Manufacturing Systems* 11(3), pp. 157 - 170.
- Cardi, A.; Firpi, H.A; Bement, M.T.; Liang, S.Y. (2008). “Workpiece dynamic analysis and prediction during chatter of turning

- process". *Mechanical Systems and Signal Processing* 22(6), pp. 1481-1494.
- Chien, W.T.; Tsai, C.S. (2003). "The investigation on the prediction of tool wear and the determination of optimum cutting conditions in machining 17-4PH stainless steel". *Journal of Material Processing Technology* 140 (1-3), pp. 340-345.
- Chua, M.S.; Loh, H.T.; Wong, Y.S.; Rahman, M. (1991). "Optimization of cutting conditions for multi-pass turning operations using sequential quadratic programming" *Journal of Materials Processing Technology* 28(1-2), pp. 253 - 262.
- Coello, C.A. (2006). "Twenty years of evolutionary multi-objective optimization: A historical view of the field". *IEEE Computational Intelligence Magazine* 1(1), pp. 28-36.
- Croitoru, C.; Severincu, M.; Belous, V. (2001). "A solution for the multi-criteria optimisation of the cutting process" [online], *Buletinul Institutului Politehnic din Isai XLVII (LI)*, pp. 83-90, [citado: 5-mar-2002], disponible en Internet en <<http://athena.mt.tuiasi.ro/icms/papers/2k1014.pdf>>.
- Cus, F.; Balic, J. (2003). "Optimization of cutting process by GA approach", *Robotics and Computer Integrated Manufacturing* 19(1-2) pp. 113-121.

- Cus, F.; Zuperl, U. (2006). "Approach to optimization of cutting conditions by using artificial neural networks". *Journal of Materials Processing Technology* 173(3), pp. 281-290.
- Dereli, T.; Filiz, I.H.; Baykasoglu, A. (2001). "Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms". *International Journal of Production Research* 39(15), pp. 3303-3328.
- Ghaiebi, H.; Solimanpur, M. (2007), "An ant algorithm for optimization of hole-making operations", *Computers and Industrial Engineering*. 52(4), pp. 308-319.
- Hasçalik, A.; Çaydas, U. (2007). "Optimization of turning parameters for surface roughness and tool life based on the Taguchi method". *International Journal of Advanced Manufacturing Technology* 38(9-10), pp. 896-903.
- Hornig, J.-T.; Liu, N.-M.; Chiang, K.-T. (2008). "Investigating the machinability evaluation of Hadfield steel in the hard turning with Al₂O₃/TiC mixed ceramic tool based on the response surface methodology". *Journal of Materials Processing Technology* 208(1-3), pp. 532- 541.
- Jeang, A.; Chung, C.-P.; Hsieh, C.-K. (2007). "Simultaneous process mean and process tolerance determination with asymmetrical loss function". *International Journal of Advanced Manufacturing Technology* 31(7-8), pp. 694-704.

- Karpat, Y.; Özel, T. (2007). "Multi-objective optimization for turning processes using neural network modeling and dynamic-neighborhood particle swarm optimization". *International Journal of Advanced Manufacturing Technology* 35(3-4), pp. 234-247.
- Kayacan, M.C.; Filiz, I.H.; Sönmez, A.I.; Baykasoglu, A.; Dereli, T. (1996). "OPPS-ROT: An optimised process planning system for rotational parts", *Computers in Industry* 32(2), pp. 181 - 195.
- Kim, S.S.; Kim, I.H.; Mani, V., and Kim, H. J. (2008). "Real-coded genetic algorithm for machining condition optimization". *International Journal of Advanced Manufacturing Technology*, 38(9-10), pp. 884-895.
- Ko, T.J.; Kim, H.S. (1998). "Autonomous cutting parameter regulation using adaptive modeling and genetic algorithms". *Precision Engineering* 22(4), pp. 243-251.
- Kumar Jain, R.; Kumar Jain, V. (2000). "Optimum selection of machining conditions in abrasive flow machining using neural networks", *Journal of Material Processing Technology* 108(1), pp. 62-67.
- Kurpati, A.; Azarm, S.; Wu, J. (2002). "Constraint handling improvement for multiobjective genetic algorithms". *Structural and Multidisciplinary Optimization* 23(3), pp. 204 - 213.
- Lee, B.Y.; Tarng, Y.S. (2000). "Cutting-parameter selection for maximizing production rate or minimizing production cost in

- multistage turning operations”. *Journal of Materials Processing Technology* 105(1-2), pp. 61-66.
- Liang, M.; Mgwatu, M.; Zuo, M. (2001). “Integration of cutting parameter selection and tool adjustment decisions for multipass turning”. *International Journal of Advanced Manufacturing Technology* 17(12), pp. 861-869.
- Meng, Q; Arsecularatne, J.A.; Mathew, P. (2000). “Calculation of optimum cutting conditions for turning operations using a machining theory” *International Journal of Machine Tools and Manufacture* 40(12), pp. 1709-1733.
- Muthukrishnan, N.; Davim, J.P. (2009). “Optimization of machining parameters of Al/SiC-MMC with ANOVA and ANN análisis”. *Journal of Materials Processing Technology* 209(1), pp. 225-232.
- Nefiódov, N. y Osípov, K. (1980). *Problemas y ejemplos de corte de metales y herramientas cortantes*, Moscú: Editorial Mir.
- Nian, C.Y.; Yang, W.S.; Tarng, Y.S. (1999). “Optimization of turning operations with multiple performance characteristics”, *Journal of Materials Processing Technology* 95(1-3), pp. 90-96.
- Ozel, C.; Kilickap, E. (2006). “Optimisation of surface roughness with GA approach in turning 15% SiC_p reinforced AlSi₇Mg₂ MMC material”. *International Journal of Machining and Machinability of Materials* 1(4), pp. 476-478.

- Özel, T. (2009). "Single and multiobjective optimization". En: Özel, T.; Davim, J.P., *Intelligent machining*, ISTE, Londres.
- Paiva, A.P.; Ferreira, J.R.; Balestrassi, P.P. (2007). "A multivariate hybrid approach applied to AISI 52100 hardened steel turning optimization". *Journal of Materials Processing Technology* 189(1-3), pp. 26-35.
- Quiza, R.; Rivas, M. (2003). "Optimización del proceso de torneado mediante un método basado en algoritmos genéticos". En: *Memorias de la 8va Convención Internacional de las Industrias Metalúrgica, Mecánica y del Reciclaje METANICA'2003* [cd-rom], La Habana, ISSN 1607-6281.
- Quiza, R.; Rivas, M.; Alfonso, E. (2006). "Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes". *Engineering Applications of Artificial Intelligence* 19(2), pp. 127-133.
- Quiza, R.; Albelo, J.E.; Davim, J.P. (2009). "Multi-objective optimisation of multipass turning by using a genetic algorithm". *International Journal of Materials and Product Research* 35(1-2), pp. 134-144.
- Raja, S.B.; Baskar, N. (2010). "Optimization techniques for machining operations: a retrospective research based on various mathematical models". *International Journal of Advanced Manufacturing Technology* 48(9-12), pp.

- Sandvik (1994). *El mecanizado moderno - manual práctico*, 1ra Edición, Sandviken (Suecia): Sandvik Coromant, Dpto. de Ediciones Técnicas, ISBN 91-972299-2-X.
- Saravanan, R.; Asokan, P.; Vijayakumar, K., 2003. "Machining parameters optimisation for turning cylindrical stock into a continuous finished profile using genetic algorithm (GA) and simulated annealing (SA)", *International Journal of Advanced Manufacturing Technology* 21(1), pp. 1-9.
- Schneider, J. J.; Kirkpatrick, S. (2006). *Stochastic optimization*, Springer-Verlag, Berlin (Germany).
- Shaw, M.C. (1984). *Metal cutting principles - Oxford series in advanced manufacturing*, Oxford (UK): Oxford University Press, ISBN 0-19-859002-4.
- Singh, G.; Choudhary, A.K.; Karunakaran, K.P.; Tiwari, M.K. (2006). "An evolutionary approach for multi-pass turning operations". *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 220(2), pp. 145-162.
- Sönmez, A.I.; Baykasoglu, A.; Dereli, T.; Filiz, I.H. (1999). "Dynamic optimization of multipass milling operation via geometric programming". *International Journal of Machine Tools & Manufacture* 39(2), pp. 287-320.
- Tang, Y. (2006). "Optimization strategy in end milling process for high speed machining of hardened die/mold steel". *Journal of University*

-
- of Science and Technology Beijing, Mineral, Metallurgy, Material*
13(3), pp. 240- 243.
- Taylor, F.W. (1907). "On the art of cutting metals". *Transactions of the ASME* 28, pp. 310 - 350.
- Tong, L.-I.; Chen, C.-C.; Wang, C.-H. (2007). "Optimization of multi-response processes using the VIKOR method". *International Journal of Advanced Manufacturing Technology* 31(11-12), pp. 1049-1057.
- Vijayakumar, K.; Prabhakaran, G.; Asokan, P.; Saravanan, R. (2003). "Optimization of multi-pass turning operations using ant colony system". *International Journal of Machine Tools and Manufacture* 43(15), pp. 1633-1639.
- Wang, J.; Kuriyagawa, T.; Wei, X.P. y Gou, G.M. (2002). "Optimization of cutting conditions using a deterministic approach". *International Journal of Machine Tools & Manufacture* 42(9), pp. 1023 - 1033.
- Wang, Y.C. (2007). "A note on 'optimization of multi-pass turning operations using ant colony system'". *International Journal of Machine Tools and Manufacture* 47(12-13), pp. 2057-2059.
- Xueping, Z.; Erwei, G.; Liu, C.R. (2009). "Optimization of process parameter of residual stresses for hard turned surfaces". *Journal of Materials Processing Technology* 209(9), pp. 4286-4291
- Yildiz, A. R.; Ozturk, F. (2006). "Hybrid enhanced genetic algorithm to select optimal machining parameters in turning operation".

- Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 220(12), pp. 2041-2053.
- Zuperl, U.; Cus, F. (2003). "Optimization of cutting conditions during cutting by using neural networks". *Robotics and Computer Integrated Manufacturing* 19(1), pp. 189 - 199.