

Un algoritmo multi-objetivo NSGA-II para la programación de los pasos de laminación en caliente de aceros

Carlos A. Hernández¹, Guadalupe Castilla², Alejandro López¹,
Juana E. Mancilla¹

¹ Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Azcapotzalco, Ciudad de México,
México

² Instituto Tecnológico de Ciudad Madero, Tamaulipas,
México

cahernandezc@ipn.mx, gpe_cas@yahoo.com.mx

Resumen. El problema de la programación de los pasos de laminación en caliente de aceros es muy complejo, ya que posee grandes espacios de búsqueda de soluciones. El objetivo de este trabajo fue la optimización multiobjetivo del proceso de laminación en caliente de láminas de acero en un tren acabador de seis pasos, implementando el *algoritmo genético de búsqueda no dominada elitista de segunda generación* (NSGA-II en inglés de *Nondominated Sorting Genetic Algorithm*), con dos cruces diferentes (AG1 y AG2) y dos funciones objetivo. Estas fueron el tiempo total de laminado y la flexión de los rodillos de trabajo. Se realizaron dos clases de experimentos computacionales: a) Búsqueda de la mejor solución, b) evaluación comparativa de las configuraciones del NSGA-II. Para cada experimento se obtuvieron 3000 soluciones. En cada molino se calcularon el tiempo de laminación, la fuerza de laminación requerida y los espesores de salida. Para la fuerza se utilizó un modelo constitutivo que calcula la tensión de fluencia, válido para cualquier tipo de acero de bajo, medio carbono y microaleados al Nb, Ti, V. Se reportan las soluciones, seleccionando la mejor de entre 100 con base en el tiempo de cómputo menor. En tiempos menores a 70 s se generaron 100 soluciones. El AG1 tuvo menor tiempo de cómputo que el AG2. Se midió el tiempo de cómputo necesario para generar cada conjunto de soluciones. El AG1 presentó el menor costo computacional. La implementación del NSGA-II minimizó tanto el tiempo de laminación como la flexión de los rodillos de trabajo, a costos computacionales bajos y asegurando que la planicidad de la lámina siempre será menor a la de estándares internacionales.

Palabras clave: Algoritmos genéticos, NSGA-II, optimización multiobjetivo, laminación en caliente, programación, pasos de laminación, aceros al carbono.

A NSGA-II Multi-Objective Algorithm for Hot Rolling Pass Schedule of Steels

Abstract. The problem of Scheduling Hot Rolling Pass of Steels is very complex because it has large spaces search for solutions. The aim of this work was the Multi-Objective Optimization (MOO) of hot rolling steel sheets in a finishing train of six steps, implementing the second generation of Nondominated Sorting Genetic Algorithm (NSGA-II). It was used two different crossover (AG1 y AG2) operators and two objective functions. These were the total time of rolling and bending of the work rolls. Two kinds of computational experiments were performed: a) Search for the best solution, b) benchmarking of both NSGA-II configurations. For each experiment was obtained 3000 solutions. Rolling time, separation roll force, and exit thickness for each rolling stand are calculated. A flow stress constitutive model for roll force was used, valid for any low, medium carbon and Nb, Ti, V microalloyed steels. The solutions are reported, choosing some of them between 100 based on the lower computing time. The 100 solutions were generated in times lower than 70s. AG1 had lower computing times AG2. The NSGA-II implementation minimized both the rolling time as crown of the work rolls, at low computational costs and ensuring that the flatness of strip will always be lower than international standards.

Keywords: Genetic algorithms, NSGA-II, multi-objective optimization, hot rolling, schedule, pass rolling, carbon steels.

1. Introducción

El proceso de laminación de aceros en caliente es uno de los métodos de conformado más utilizados para la obtención de productos para diferentes industrias metalmeccánicas. En este proceso, el espesor de un planchón de acero se reduce gradualmente aplicando fuerzas de compresión mediante rodillos de laminación [1]. Las fuerzas de laminación aplicadas tienden a incrementarse a medida que el porcentaje de reducción del espesor de la lámina aumenta, pudiendo llegar a provocar flexión en los rodillos de trabajo, pudiendo provocar imperfecciones en la superficie de la lámina o defectos de planicidad por corona de flexión. Esto último produce una lámina que es más delgada en el centro que en las orillas [2]. Existen métodos mecánicos para corregir este tipo de problemas, como el rectificando de los rodillos. Así, cuando el rodillo se somete a fuerzas de separación y se flexiona se obtendrá un espesor constante en la lámina. En el caso de laminación de aceros, el radio del punto máximo de la comba es en general 0.25 mm mayor que en los extremos de los rodillos [3]. Esto implica que el rodillo sólo puede someterse a una carga específica, lo cual en la industria es ineficiente. De no ser así, las láminas producidas por este rodillo presentarían defectos de

planicidad. Además, una mala selección de los pasos de laminación incrementa el tiempo del proceso, lo cual es improductivo.

Cuando la solución de un problema tiene un espacio de búsqueda muy amplio, existen limitaciones de tiempo para llegar a una solución útil y no existe un método analítico mediante el cual se pueda llegar a una solución exacta. Por ello se usan técnicas metaheurísticas para encontrar soluciones óptimas alejadas de óptimos locales; es decir, se orienta la búsqueda acorde a la evolución del proceso. Los algoritmos evolutivos utilizan mecanismos inspirados en la evolución biológica, tales como la selección, reproducción y la mutación. Las soluciones se modelan como individuos de una población. Una función de aptitud determina el entorno donde se encuentran las soluciones. Dentro de los algoritmos evolutivos están los algoritmos genéticos (AG), que son una técnica de búsqueda iterativa, buscando derivar estrategias de optimización basadas en la generación de poblaciones de individuos aptos mediante la reproducción de los padres [4]. Los problemas reales usualmente requieren la búsqueda de soluciones que satisfagan en forma simultánea múltiples criterios de desempeño u objetivos que pueden ser contradictorios, es decir, se desconoce la manera óptima de combinar los diferentes objetivos o que sea inadecuado, cuando no es imposible hacerlo, a lo cual se le conoce como un Problema de Optimización Multiobjetivo (MOP en inglés de *Multiobjective Optimization Problem*) [5].

Ya que el proceso de laminación en caliente obedece a más de una variable, el espacio de soluciones óptimas puede ser infinito y el tiempo para obtener solo una de ellas con métodos analíticos puede tomar demasiado tiempo y disminuir la eficiencia del proceso. Por ello, la aplicación de técnicas metaheurísticas a este proceso ha sido objeto de varias investigaciones cuyos autores han enfocado su atención en diferentes variables. En este trabajo se trata de encontrar los espesores intermedios óptimos (h_1, \dots, h_{f-1}) tales que minimicen tanto el tiempo total de laminación como la flexión máxima de los rodillos de trabajo. Para la flexión, esta deberá ser menor a los límites de planicidad establecidos por algunas normas internacionales como la EN 10 051, ASTM A568 o A653, la JIS G-3193 o G-3116 y la DIN 10051. Por tratarse de una investigación de carácter multiobjetivo, este trabajo se integra a las investigaciones cuyos propósitos son la optimización de las variables intrínsecas del proceso, para reducir el tiempo total de laminación, así como la optimización de la operación y vida útil de la maquinaria para disminuir la máxima flexión de los rodillos de trabajo de la cédula de laminación. Por ello, en las últimas dos décadas, el proceso de laminación ha sido objeto de estudio aplicando diferentes técnicas metaheurísticas y, más recientemente, de algoritmos evolutivos para la optimización multiobjetivo del proceso, por el hecho de que existe más de una restricción, algunas de las cuales se mencionan a continuación.

Nolle y Armstrong [3] realizaron la modelización del proceso de un tren acabador de laminación de siete pasos aplicando redes neuronales [6], dirigido a la optimización de factores cuyos parámetros se reflejaban en la planicidad y la deformación longitudinal de una lámina obtenida mediante un tren laminador. Un desarrollo importante en el campo de la aplicación de los algoritmos

genéticos (AG) a la laminación en frío fue propuesto por Wang y cols. [7] quienes optimizaron la programación de una cédula de laminación en frío. En su investigación obtuvieron los parámetros que optimizaban el proceso infiriendo el incremento en la calidad de la lámina y el aumento de la vida útil de los rodillos de laminación. Wang demostró que los parámetros se mejoraron destacando la reducción de la potencia empleada, las revoluciones por minuto necesarias para el proceso, así como el par requerido. En 2003 Nolle y Zelinka [8] desarrollaron un algoritmo heurístico de optimización denominado algoritmo de migración auto organizado (SOMA del inglés *Self-Organizing Migration Algorithm*). Tanto el objetivo de ese trabajo como la función a optimizar fueron los mismos del trabajo publicado en 1999 [3]. SOMA se aplicó en un programa de laminación para catorce diferentes láminas y se relacionó con la calidad de las láminas obtenidas por el proceso de laminación. Los resultados arrojados se compararon con los de la planicidad de láminas medidas en *I-Units* provenientes de pruebas industriales, obteniendo SOMA un 33% de mejora. Estos autores también desarrollaron un AG para la optimización mediante simulación de los rodillos de trabajo del tren acabador. El factor a optimizar fue una función compuesta por una combinación de parámetros de planicidad. Se comparó el AG con ocho diferentes algoritmos de optimización experimentales, como lo son el recocido simulado, ascenso de colina y la trayectoria aleatoria. Demostraron que el AG fue el que obtuvo los mejores resultados. En 2007, Nolle [9] implementó un algoritmo heurístico de búsqueda de tamaño de pasos auto adaptable (SASS del inglés *self-adaptive step size search*), para optimizar el perfil de los rodillos de trabajo utilizados en el proceso de laminación en caliente. El SASS es un algoritmo basado en los principios aplicados en la técnica de optimización experimental de ascenso de colina, donde únicamente se puede configurar el tamaño de la población y aunque el algoritmo es muy simple, los resultados obtenidos fueron muy buenos. Los resultados arrojados por SASS se valoraron en relación con la planicidad de la lámina y su deformación longitudinal, siendo mejores que los datos industriales contra los cuales se compararon. Recientemente, Pholdee et al. [10] optimizó la planicidad de láminas durante el bobinado utilizando un método de evolución diferencial junto con varios algoritmos evolucionarios. Este método realiza gran cantidad de iteraciones, por lo cual requiere nuevos algoritmos evolucionarios que reduzcan la cantidad de evaluaciones a las funciones.

2. Formulación del problema de optimización

El problema de la optimización de la programación de los pasos de laminación en caliente para la fabricación de lámina plana radica en que la minimización del tiempo de procesamiento no toma en cuenta el efecto que tendría sobre la flexión del rodillo de trabajo y con ello la afectación del perfil de la lámina, tal que sobrepase las tolerancias cada vez más exigentes para la calidad de las láminas. Ya que las minimizaciones del tiempo de laminación y de la flexión del rodillo involucran soluciones en conflicto se requiere resolver el problema de optimización mediante algoritmos multiobjetivos.

Existen muchos métodos de solución para encontrar un conjunto de soluciones lo más cercano posible a un frente óptimo de Pareto de los tiempos de laminación t y la flexión del rodillo ν , en cada paso de laminación. Ya que las fuerzas de laminación están gobernadas por los parámetros del proceso y por la resistencia a la deformación del material, conviene que el tiempo de laminación y la flexión del rodillo óptimos provengan de un NSGA-II, ya que permite hallar un número de soluciones no dominadas que convergen en un frente óptimo de Pareto. Para este trabajo se han considerado tres restricciones, a) la potencia óptima de laminación ($P_{\text{ópt}}$) para obtener un espesor de salida (h_i) sea menor que la potencia nominal (P_{nominal}) de cada molino de laminación, b) el espesor de salida (h_i) sea estrictamente menor que el espesor de entrada (h_{i-1}) y c) la corona de flexión ($\nu_{\text{ópt}}$) sea menor a la establecida por las normas de planicidad (ν_{norma}). Entonces el problema de optimización se formula como

$$\left. \begin{array}{l} \text{minimizar } f_1(t, \nu) = \sum_{n=1}^n t_i, \\ \text{minimizar } f_2(t, \nu) = \frac{R_A x^3}{6EI} - \frac{q(x - l_1)^4}{24EI} + \left(\frac{qx(\frac{L}{2} - l_1)^3}{6EI} - \frac{R_A x L^2}{8EI} \right), \\ \text{sujeto a: } P_{\text{ópt}} < P_{\text{nominal}} \\ \quad \quad h_{i-1} < h_i \\ \quad \quad \nu_{\text{ópt}} < \nu_{\text{norma}}, \end{array} \right\} \quad (1)$$

siendo R_A la reacción en un extremo del rodillo (punto A), x la distancia longitudinal dada a lo largo de la cual ocurre la flexión del rodillo, l_1 la longitud de la primera sección del rodillo de trabajo, L la longitud total del rodillo, E el módulo de elasticidad del material del rodillo e I el momento de inercia de la sección transversal del rodillo. Los parámetros de entrada del planchón se pueden consultar en trabajos previos [11,12].

3. Parámetros del AG

El algoritmo genético se inicializa mediante parámetros establecidos por el usuario previamente (Tabla 1).

Tabla 1. Parámetros del NSGA-II.

Parámetro	Magnitud
Población inicial	100
Número de generaciones	100
Taza de cruzamiento	0.9 %
Taza de mutación	0.2 %

4. Instancias empleadas

Las instancias para la experimentación en este trabajo se obtuvieron de trabajos previos [13]. Para determinar la funcionalidad del AG desarrollado, así como sus variantes en el tipo de cruza, se probaron 17 instancias. Cada instancia representa un problema diferente. En cada problema se determinaron las reducciones intermedias para laminar un acero hasta obtener el espesor final deseado en un tren de laminación de 6 pasos. Cada instancia contiene el espesor inicial del planchón (h_0), el espesor final (h_f) deseado, la composición química del material (%C, %Mn, %Si, %Mo, %Nb, %Ti, %V) y el ancho del planchón (w). También, y para cada paso del tren de laminación, se deben especificar diámetros de rodillos de trabajo (D_i), velocidades de rodillos (v_i), distancias entre las cajas de laminación (l_i), potencias de motores (P_i), temperaturas intermedias (T_i) y el tamaño inicial de grano (d_{0_i}).

5. Solución al problema de optimización

Para dar solución al problema de optimización de una cédula de laminación en caliente en esta investigación se empleó el NSGA-II. Se configuró al AG en dos formas diferentes. En la primera se utilizó la cruza en dos puntos (AG1) y en la segunda se utiliza cruza SBX (AG2). En ambos casos se implementó una selección aleatoria elitista para generar la población inicial y se mantuvo una mutación constante.

La población inicial está compuesta sólo por individuos factibles, que evolucionan en busca de la mejor solución al problema. El número de individuos lo define el usuario al inicio de la ejecución del algoritmo, al igual que el número de generaciones, que representa el ciclo del algoritmo.

5.1. Algoritmo NSGA-II

En el Algoritmo 1 se muestra el código básico del NSGA-II. Para que se genere la primera población de padres y cumpla su ciclo de operación, los individuos deben cumplir con todas las restricciones del problema.

En el Algoritmo 2 se muestran las operaciones más importantes, así como los cálculos para determinar las restricciones del problema con las cuales se clasifica a un individuo como viable o no viable.

5.2. Generación población inicial

Dado un espesor inicial h_0 y un espesor final h_f del planchón, considerados constantes del problema, se generan individuos cuya representación es la siguiente:

$$X = [h_0, h_1, h_2, h_3, h_4, h_5, h_f] . \quad (2)$$

A continuación se explica brevemente la solución obtenida para la instancia número 1. Con los datos de la instancia y mediante la ecuación (3), que representa

Algoritmo 1 Ciclo principal de operaciones de NSGA-II.

```

1: Inicio
2: Repite.
3: Inicialización de parámetros
4: Genera individuos aleatorios  $\mu$ .
5: Evalúa a los individuos.
6: Si  $\mu$  es factible entonces
7:      $\mu \rightarrow P_t$ 
8:     Deshecha  $\mu$ 
9:     Hasta Criterios de término.
10:  $F$ =Ordenamiento no dominado ( $P_t$ )
11: Genera  $Q_t$  a partir de  $P_t$ 
12: Mientras (Criterio de parada) Repite
13:     Genera  $R_t = Q_t \cup P_t$ 
14:      $F$ =Ordenamiento no dominado rápido ( $R_t$ )
15:     Mientras  $|P_{t+1}| \geq N$  hacer
16:         Distancia de Agrupamiento ( $F_i$ )
17:          $P_{t+1} = P_{t+1} \cup F_i$ 
18:         Ordena ( $P_{t+1}, \geq n$ )
19:          $P_{t+1} = P_{t+1}[0 : N]$ 
20:          $Q_{t+1}$ =Genera nueva población  $P_{t+1}$ 
21:          $t = t + 1$ 
22:     Fin Mientras
23: Fin

```

la reducción media \bar{r} para n pasos de laminación (=6), se obtienen los individuos de la población inicial:

$$\bar{r} = 1 - \sqrt[n]{\frac{h_f}{h_0}} . \quad (3)$$

Cada individuo representa una solución al problema:

$$X = [48, 36, 25, 18, 10, 6, 38] ,$$

e implica que tal individuo es factible al haber cumplido con las restricciones del problema.

La población de 100 individuos generada representa a la población inicial (P_0), que se clasifica según los niveles de no dominancia en frentes de Pareto (F_1, F_2, \dots, F_n). Una vez que todos los individuos se han ordenado, se les asignó una calificación denominada aptitud. Los individuos que ocupan el frente F_1 reciben la mejor calificación, debido a que son individuos no dominados, como se muestra en la Tabla 2.

5.3. Cruza

Se considera a cada individuo como la representación de una solución para el problema y cada solución tiene un rango de no dominancia asociado (f_i), así como una distancia de apilamiento (d_i).

Algoritmo 2 Ciclo de operaciones principales para determinar si un individuo satisface las restricciones del problema.

-
- 1: **Inicio**
 - 2: **Entrada** Especificaciones del planchón, h_0, h_f, T_i, d_0
Composición química %C, %Mn, %Si, %Mo, %Nb, %V no vacío.
 - 3: **Entrada** Especificaciones de la cédula de laminación, l_i, P_i, v_i, n no vacío.
 - 4: **Entrada** Especificaciones de los rodillos de trabajo, D_i, L_i, E .
 - 5: **Entrada** Restricciones X_n
 - 6: **Salida Individuo viable** si X_n se cumplen
 - 7: **Individuo no viable** en caso contrario.
 - 8: **Repite.**
 - 9: Genera individuos aleatorios μ_i .
 - 10: **Para** μ_i **Calcula**
 - 11: Tensión de Fluencia σ_i
 - 12: Resistencia a la deformación $\bar{\kappa}_i$
 - 13: Fuerza de laminación F_i
 - 14: Calcula par de laminación G_i
 - 15: Calcula potencia de laminación σ_i
 - 16: Calcula flexión en los rodillos de trabajo $Flex_i$.
 - 17: Calcula tiempo total de laminación t_{total} .
 - 18: Evalúa a los individuos.
 - 19: **Si** μ_i es factible
 - 20: $\mu \rightarrow P_t$
 - 21: **Si no** Descarta μ_i
 - 22: **Hasta** Criterios de término.
 - 23: **Fin**
-

Tabla 2. Ordenamiento en frentes no dominados de Pareto.

Frente	Padres	Fitness
F_1	2, 8, 9, 5	10
F_2	1, 3, 4	9
F_3	7, 6	8
F_4	...	7
F_n
\sum Padres = 100		

Una vez completado el ordenamiento de la población inicial de padres P_0 en los frentes de Pareto se procedió a su cruce mediante la técnica de torneo binario. Esta comienza con la selección aleatoria de dos individuos de la población de padres. La mejor solución es por comparación y el ganador es el que tiene la mejor calificación o rango de no dominancia. En el caso de estar situadas en el mismo frente de Pareto, la ganadora del torneo es aquella que tiene mejor distancia de apilamiento, también conocida como distancia de Crowding d_i , que representa la mayor distancia entre las soluciones de la población asegurando la diversidad de soluciones.

La d_i se calcula mediante la ecuación (4),

$$d_j^m = \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{\max} - f_m^{\min}}, \quad (4)$$

siendo f_m^{\max}, f_m^{\min} los valores máximo y mínimo de la función objetivo m y $f_m^{I_{j+1}^m}, f_m^{I_{j-1}^m}$ son las soluciones vecinas a la j -ésima configuración para cada una de las funciones objetivo m . La distancia resultante fue la suma de las distancias en cada una de las direcciones de las funciones objetivo. Se dio preferencia a aquellos individuos localizados en una región menos poblada para generar dispersión en las soluciones.

Una vez que los padres ganadores del torneo binario se seleccionaron, se realizó la cruce entre ellos para generar una población de tamaño N . Los métodos de cruce fueron la de dos puntos y SBX. Para el cálculo de la primera generación del hijo₁, sólo se consideran los h_1 de ambos padres. Aplicando la ecuación para la cruce SBX, cuando $\beta = 1.5$, $P_1 = 0.032$ m y $P_2 = 0.034$ m se tiene que:

$$\begin{aligned} H_1 &= 0.5[(P_1 + P_2 - \beta(P_1 - P_2))] = 0.0315, \\ H_2 &= 0.5[(P_1 + P_2 + \beta(P_1 - P_2))] = 0.0345. \end{aligned}$$

5.4. Mutación

El operador mutación es necesario para preservar la diversidad genética de la población evitando divergencia prematura. En este trabajo se implementó mutación uniforme para el desarrollo del NSGA-II. La forma en que se desarrolla este tipo de mutación es la siguiente:

$$\begin{aligned} P &= [h_0, h_1, h_f], \\ P' &= [h_0, h'_1, h_f], \end{aligned}$$

siendo $h'_1 = \text{random}(LI, LS)$. Se utilizó una distribución uniforme y (LI, LS) definen los límites mínimos y máximos de la variable h'_1 .

5.5. Unión padres e hijos

En este algoritmo, la población descendiente Q_t de tamaño N se crea en primera instancia utilizando la población de padres P_t de tamaño N . Después, las dos poblaciones se combinan para formar R_t de tamaño $2N$. A esta nueva población $R_t = P_t \cup Q_t$ se le deberán aplicar todas las operaciones para generar la nueva población de individuos. Sin embargo, como la población R_t es de tamaño $2N$ y solamente existen N configuraciones que conforman la población descendiente, no todas las configuraciones de los frentes pertenecientes a la población R_t podrán integrarse a la nueva población, por lo cual aquellos frentes

que no pueden ser acomodados desaparecen, es decir no son considerados en las nuevas poblaciones, asegurando así que sólo los frentes más aptos sigan reproduciéndose.

El ciclo de operaciones del NSGA-II es repetitivo en todas las generaciones, por lo cual sucesivamente se crea la población descendiente Q_{i+1} , a partir de P_{i+1} , aplicando selección por torneo para apilamiento, cruza y mutación, hasta que la instrucción de paro del algoritmo se cumpla. La instrucción de paro es parte de los parámetros de entrada del algoritmo.

6. Evaluación comparativa

Debido a que se implementó el NSGA-II configurado en dos distintas formas, modificando la técnica de cruza en cada uno de ellos, es conveniente establecer si el tipo de configuración afecta los resultados obtenidos. El objetivo de la evaluación comparativa fue identificar si existen diferencias en los resultados obtenidos de las diferentes configuraciones del NSGA-II.

En la Tabla 3 se muestran las dos configuraciones utilizadas en este trabajo.

Tabla 3. Configuraciones del NSGA-II para este estudio.

Algoritmo	Tipo de selección	Cruza	Mutación
AG1	E. por torneo	2 puntos	Uniforme
AG2	E. por torneo	SBX	Uniforme

Se consideró al costo computacional para propósitos de la evaluación. Para realmente lograr una comparación objetiva entre soluciones, es necesario definir una medida cuantitativa de la calidad de las mismas. Para problemas de optimización multiobjetivo donde no existe una solución única sino un conjunto de soluciones, no es fácil definir un criterio para determinar cuándo un conjunto es mejor que otro debido a que se espera que un conjunto de soluciones se encuentre compuesto únicamente por soluciones mutuamente no comparables. Es decir, para cualquier par de soluciones \vec{x}_1, \vec{x}_2 no ocurre que $\vec{x}_1 \leq \vec{x}_2$ ni $\vec{x}_1 \geq \vec{x}_2$ debido a que todas ellas son viables y es el usuario quien debe seleccionar una de modo manual o programado.

Es posible definir un indicador de calidad que asigna un valor real a un conjunto de aproximación del frente de Pareto. Un indicador debe definir un orden total en los conjuntos de aproximación del frente de Pareto. Si se tienen dos conjuntos de aproximación A, B tales que $I(A) > I(B)$, entonces el conjunto A es preferible sobre B . Se puede asignar un indicador particular. Por tal motivo puede ocurrir que si $I(A) > I(B)$ utilizando algún indicador, entonces $I'(A) > I'(B)$ tendrá un indicador distinto.

7. Condiciones de la experimentación

Las condiciones fueron las siguientes:

- Se empleó una computadora con procesador Intel Core *i7* con una velocidad máxima de procesamiento de 3.1 GHz y 12 GB RAM.
- Se realizaron 30 corridas por cada una de las 17 instancias para obtener los valores promedios de cada una de ellas así como garantizar la fiabilidad de los resultados. Cada corrida contiene en un archivo de salida con 100 soluciones, todas viables porque cumplieron con las restricciones del problema.

Se implementó el algoritmo bajo los parámetros mostrados en la Tabla 4.

Tabla 4. Parámetros del NSGA-II.

Parámetro	Magnitud
Población inicial	100 ind.
Núm. generaciones	100 gen.
Cruza	95%
Mutación	20%

Para cada experimento se obtuvieron tres mil soluciones que satisfacen a las funciones objetivo de este problema. Además, en cada molino se obtuvo el tiempo de laminación, la fuerza de laminación y los espesores intermedios óptimos. Así mismo se obtuvieron los tiempos de cómputo de cada experimento.

8. Búsqueda de la mejor solución

En otras propuestas de algoritmos publicados en la literatura para el problema de laminación en caliente, el espacio de soluciones quedó restringido a la búsqueda de únicamente una solución por cada experimento que entregan buenos resultados, pero suelen tener altos tiempos de cómputo. Las soluciones obtenidas en este trabajo mediante el AG sirvieron para seleccionar la mejor de ellas. Esta depende directamente de los requerimientos del usuario, es decir es él quien selecciona una respuesta del conjunto como la mejor, basado en su experiencia y conocimiento del proceso, con la seguridad de que cualquiera de ellas cumple con las restricciones del problema. En esta investigación se seleccionó aquella corrida que generó el menor tiempo de cómputo, evaluándose 100 soluciones. Se consideraron dos indicadores principales, tiempo de laminación y flexiones máximas en los rodillos de trabajo; se dio un peso específico a cada indicador. El indicador con mayor peso es el tiempo de laminación. Por ser un problema de minimización se seleccionó como mejor solución la que presentó aquellos indicadores cuyos valores son los mínimos. Con base en este criterio se obtuvieron los siguientes resultados.

8.1. Tiempo de cómputo

En la Fig. 1 se muestra el tiempo de cómputo promedio que le tomó a cada una de las configuraciones del algoritmo propuesto en generar una corrida para la instancia 1. En un lapso de 15 a 70 segundos, el AG es capaz de generar conjuntos de 100 soluciones.



Fig. 1. Tiempo promedio para generar una corrida de la Instancia 1.

Nótese que el AG1 tiene una mejor eficiencia sobre el AG2 pues en todos los casos requiere menor tiempo de cómputo para generar una corrida. Además, para el proceso de selección de la mejor solución de nuestro problema se eligió la corrida que requirió de menor tiempo de cómputo, enmarcadas con circunferencias en dicha Fig. 1. La corrida generada en menor tiempo fue la 14 para el AG1 y la dos para el AG2.

También se seleccionó de entre las 100 soluciones aquella que mejor se adaptó al problema. En la Fig. 2 se muestran los tiempos de laminación para cada una de las 100 soluciones determinadas mediante el AG1, siendo la solución 71 la que tuvo el menor tiempo de laminación.

En la Fig. 3 se muestran los resultados para tiempos de laminación obtenidos mediante el AG2, para la instancia número 1, apreciándose la similitud con los resultados obtenidos mediante el AG1. Sin embargo el tiempo mínimo de laminación obtenido mediante el AG2 es mayor que el del AG1. Se mejoró hasta en un 0.37% el tiempo de laminación industrial.

9. Evaluación comparativa entre las diferentes configuraciones del NSGA-II

Parte fundamental de este trabajo fue determinar el costo computacional de cada una de las variantes del algoritmo propuestas, es decir, el tiempo de cómputo necesario para generar cada conjunto de soluciones. El costo computacional



Fig. 2. Tiempos de laminación para la instancia número 1 producidos por el AG1. Con círculo azul la que tuvo el menor tiempo de laminación.



Fig. 3. Tiempos de laminación para la instancia número 1 producidos por el AG2.

se midió desde que inicia la puesta en marcha del algoritmo hasta culminar con la generación del archivo de salida que contiene las soluciones. Ya que el costo computacional es un parámetro utilizado para determinar la mejor solución, fue necesario cronometrarlo.

En la Fig. 4 se muestran los resultados obtenidos aplicando las distintas configuraciones del NSGA-II. Se observa que el AG2, representado en color azul, presentó en todas las instancias mayor costo computacional, es decir tardó mayor tiempo en generar un conjunto de soluciones. Esto se debe en gran medida a que para realizar la cruce SBX es necesario que el algoritmo resuelva ecuaciones para determinar a las nuevas generaciones, a diferencia de la cruce de dos puntos, que emplea un sistema de generación de individuos menos complejo. La diferencia entre los tiempos de cómputo obtenidos en cada una de las corridas aplicando ambas configuraciones del NSGA-II fue muy pequeña, así como en la calidad de la solución obtenida.

En la Fig. 5 se muestra el frente de Pareto con las dos funciones objetivo para el molino 6. Nótese que en la flexión que sufren los rodillos cuando se aplican los parámetros de una cédula de laminación industrial superó los límites establecidos de flexión bajo las normas de planicidad. Por otra parte, las soluciones en el

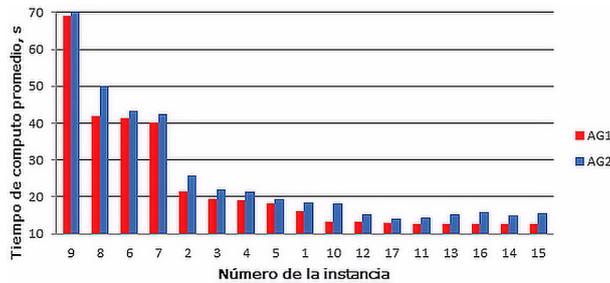


Fig. 4. Comparación entre los tiempos que tarda el AG1 y el AG2 en determinar un conjunto de 100 soluciones.

frente de Pareto encontradas mediante el AG propuesto en todas las ocasiones se mantienen bajo las tolerancias permisibles.

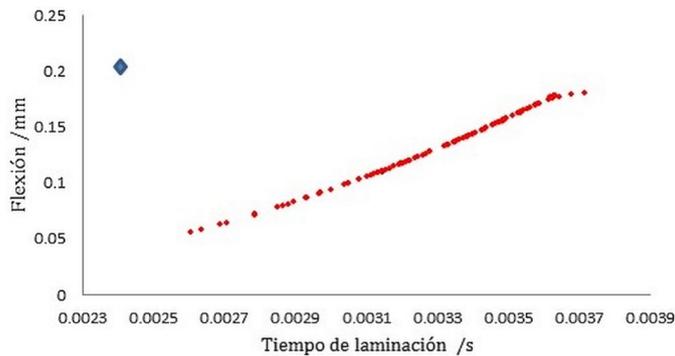


Fig. 5. Frente de Pareto para las flexiones y tiempos calculados mediante el AG (círculos rojos) en comparación con datos industriales (rombo azul).

En la Fig. 6 se muestran las reducciones determinadas por el algoritmo implementado, comparadas con las reducciones reportadas en la cédula de laminación industrial. Se puede decir que las reducciones determinadas mediante el algoritmo están mejor distribuidas en los pasos de laminación, para generar reducciones menos críticas que disminuyan la flexión en los rodillos y la potencia de laminación necesaria. Asimismo, se puede observar que los primeros tres pasos de laminación reducen a más de dos terceras partes el espesor inicial del planchón y los últimos tres pasos son más suaves, ya a que los pasos finales son pasos para dar el acabado que requiere la lámina.

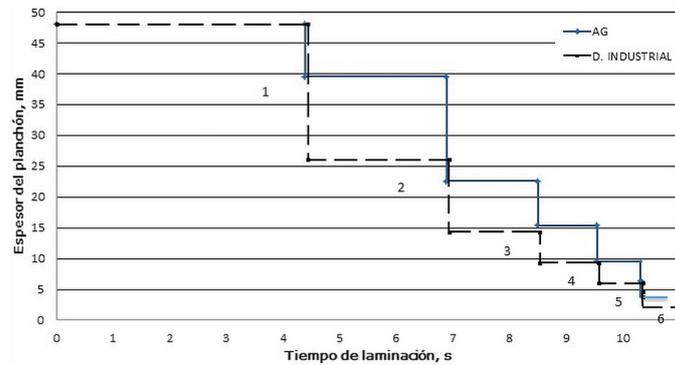


Fig. 6. Comparación entre la cédula obtenida por el algoritmo genético y la cédula industrial. Los números indican el paso de laminación.

10. Conclusiones

- En este trabajo se implementó el NSGA-II para determinar un conjunto de 100 soluciones a un bajo costo computacional.
- La implementación del NSGA-II con sus variantes AG1 y AG2 minimizaron la flexión de los rodillos de trabajo con respecto a la obtenida en datos industriales y mejoró en un 0.37 % el tiempo de laminación industrial.
- La minimización de la corona de flexión asegura que la planicidad de la lámina siempre fue menor a la establecida en normas internacionales.
- Se demostró la factibilidad de la aplicación de algoritmos genéticos multiobjetivo a esta clase de problemas por su fácil adaptación a modelos matemáticos complejos, así como al bajo costo computacional.

Agradecimientos. Este trabajo fue realizado con fondos del Instituto Politécnico Nacional (proyectos clave SIP201612163 y 20161930). Se agradece a Richard Shulkosky de la compañía INTEG Process Group por proporcionar una versión demo del simulador HSMM.

Referencias

1. Petty, D.M.: Application of process modelling—an industrial view. *J Mater Proc Technol* 60, pp. 421–426 (1996)
2. Tseng, A.A., Tong, S.X., Chen, T.C.: Thermal expansion and crown evaluations in rolling processes. *Mater Design* 17, pp. 193–204 (1996)
3. Nolle, L., Armstrong, A., Hopgood, A., Ware, A.: Optimum Work Roll Profile Selection in the Hot Rolling of Wide Steel Strip Using Computational Intelligence. In: *Computational Intelligence: Theory and Applications International Conference*, Dortmund, Germany, Reusch, B. (eds). Springer, Berlin (1999)

4. Holland, J.H.: The dynamics of searchers directed by genetic algorithms. En Lee, Y.C. (Ed.), *Evolution, learning, and cognition*, Teaneck, N. J. World Scientific (1988)
5. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8, pp. 173–195 (2000)
6. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction*. pp. 209–218, Wiley, New York (2001)
7. Wang, D.D., Tieu, A.K., de Boer, F.G., Ma, B., Yuen, W.Y.D.: Toward a heuristic optimum design of rolling schedules for tandem cold rolling mills. *Eng Appl Artificial Intell* 13(4), pp. 397–406 (2000)
8. Nolle, L., Zelinka, I.: SOMA applied to Optimum Work Roll Profile Selection in the hot rolling of wide steel. In: *17th European Simulation Multiconference*, Nottingham, UK (2003)
9. Nolle, L.: SASS applied to optimum work roll profile selection in the hot rolling of wide steel. *Knowledge-Based Systems* 20(2), pp. 203–208 (2007)
10. Pholdee, N., Bureerat, S., Park, W.W., Kim, D.K., Im, Y.T., Kwon, H.C., Chun, M.S.: Optimization of Flatness of Strip during Coiling Process based on Evolutionary Algorithms. *Int J Precis Eng Man* 16(7), pp. 1493–1499 (2015)
11. Hernández Carreón, C.A., Fraire Huacuja, H.J., Espriella Fernandez, K., Valdez Castilla, G., Mancilla Tolama, J.E.: Application of Genetic Algorithms to Strip Hot Rolling Scheduling. In: Corchado, E. y col. (eds.) *Innovations in Hybrid Intelligent Systems*, vol. 44. *Advances in Soft Computing*, pp. 247–254. Springer, Berlin (2007)
12. Hernández Carreón, C.A., Fraire-Huacuja, H.J., Fernandez, K.E., Castilla-Valdez, G., Mancilla Tolama, J.E.: A Model to Minimize the Hot Rolling Time of a Steel Slab Considering the Steel Chemical Composition. In: Corchado, J.M. y col. (eds.) *International Symposium on Distributed Computing and Artificial Intelligence 2008*, vol. 50. *Advances in Soft Computing*, pp. 471–480. Springer, Berlin (2009)
13. Espriella, K: *Optimización mediante algoritmos genéticos: aplicación a la laminación en caliente*. Tesis de Maestría, Instituto Politécnico Nacional, CICATA Unidad Altamira (2008)