



Lámpsakos

ISSN: 2145-4086

Universidad Católica Luis Amigó

Arango Marín, Jaime Antero; Rosero Otero, Silvio León; Montoya Arias, Mario Enrique  
Programación de mantenimiento preventivo usando algoritmos genéticos  
Lámpsakos, núm. 23, 2020, Enero-Junio, pp. 37-44  
Universidad Católica Luis Amigó

DOI: <https://doi.org/10.21501/21454086.3112>

Disponible en: <https://www.redalyc.org/articulo.oa?id=613964510003>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org  
UAEM

Sistema de Información Científica Redalyc  
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



# Programación de mantenimiento preventivo usando algoritmos genéticos

## Preventive maintenance programming using genetic algorithms

**Jaime Antero Arango Marín\***

**Silvio León Rosero Otero\*\***

**Mario Enrique Montoya Arias\*\*\***

(Recibido el 18-02-2019. Aprobado el 16-01-2020)

**Estilo de citación de artículo:**

J. A. Arango Marín, S. L. Rosero Otero, y M. E. Montoya Arias, "Programación de mantenimiento preventivo usando algoritmos genéticos", *Lámpsakos*, (23), pp. 37-44. (enero-junio, 2020). DOI: [10.21501/21454086.3112](https://doi.org/10.21501/21454086.3112)

### Resumen

En un sistema de mantenimiento preventivo programado por tiempo, se plantea la necesidad de optimizar el uso de los recursos del área de mantenimiento, para buscar un equilibrio a lo largo del periodo de planificación. El problema se refiere a la secuenciación de tareas en la que se requiere determinar la semana inicial del programa de mantenimiento para cada una de las actividades en las máquinas. Se propone un algoritmo genético que tiene como función objetivo minimizar el máximo tiempo de mantenimiento semanal, de tal manera que cumpla con las condiciones de la programación establecida en cuanto a duraciones y frecuencias. Se consigue un programa eficiente y equilibrado que mejora significativamente el uso de los recursos. Para futuros trabajos, se sugiere aplicar otras técnicas inteligentes para la solución del problema y formularlo teniendo en cuenta otras condiciones, como los diferentes tipos de mantenimiento o la minimización de las paradas frecuentes de los equipos.

**Palabras clave:** Mantenimiento preventivo; Mantenimiento predictivo; Algoritmos genéticos; Programación de tareas; Naturaleza combinatorial; cromosoma; Mutación; Metaheurística; Eficiencia; Programación óptima.

\* Ph.D en Ingeniería Industrial y Organizaciones. Integrante del grupo de investigación en Desarrollos Tecnológicos y Ambientales – GIDTA, Universidad Católica de Manizales, Manizales-Colombia. Contacto: [jaarangom@unal.edu.co](mailto:jaarangom@unal.edu.co), Orcid: <https://orcid.org/0000-0003-0626-6013>

\*\* MSc en Gerencia del Talento Humano. Integrante del grupo de investigación en Desarrollos Tecnológicos y Ambientales – GIDTA, Universidad Católica de Manizales, Manizales-Colombia. Contacto: [srosero@ucm.edu.co](mailto:srosero@ucm.edu.co), Orcid: <https://orcid.org/0000-0002-4290-648X>

\*\*\* Especialista en Planeación para la Educación Ambiental. Integrante del grupo de investigación en Desarrollos Tecnológicos y Ambientales – GIDTA, Universidad Católica de Manizales, Manizales-Colombia. Contacto: [mmontoya@ucm.edu.co](mailto:mmontoya@ucm.edu.co), Orcid: <https://orcid.org/0000-0003-4327-9382>

## Abstract

In a preventive maintenance system programmed by time the need to optimize the use of the resources of the maintenance area, looking for a balance throughout the planning period. The problem relates to the sequencing of tasks in which it is necessary to determine the initial week of the maintenance programme for each of the activities on the machines. A genetic algorithm is proposed that has as its objective the minimization of the maximum weekly maintenance time, in such a way that it meets the conditions of the established programming in terms of durations and frequencies. The result is an efficient and balanced programme, that significantly improves the use of resources. For future work, it is suggested to apply other intelligent techniques to the solution of the problem and to formulate it taking into account other conditions such as the different types of maintenance or the minimisation of frequent equipment stops.

**Keywords:** Preventive maintenance; Genetic algorithms; Maintenance task; Combinatorial nature; Metaheuristic; Optimal programming.

## 1. INTRODUCCIÓN

En los entornos productivos se busca asegurar la máxima disponibilidad de la maquinaria, equipo e instrumental necesarios para cumplir con las diferentes operaciones. Para evitar sobrecostos, interrupciones inesperadas en los procesos y prolongar la vida útil de las máquinas, equipos e instrumentos se hace necesario gestionar las operaciones de mantenimiento. Algunos conceptos comunes para el manejo del mantenimiento son:

### Mantenimiento correctivo

Es el que se debe efectuar para volver a poner en marcha la maquinaria y equipo que ha fallado por algún daño o avería. Generalmente implica la suspensión de actividades productivas y la reprogramación o reprocesamiento de parte de los trabajos programados [1].

### Mantenimiento preventivo

Responde a un programa preconcebido, basado en un conjunto de actividades programadas de acuerdo con un cronograma que, según el cumplimiento de fechas calendario, horas, días de operación o unidades procesadas, establece las actividades de mantenimiento a efectuar a cada una de las máquinas, equipos o instrumentos. Las suspensiones de operación son mínimas y están previstas con anticipación. Así mismo, es mínimo el reproceso o reprogramación de trabajos. Se prevén los recursos necesarios en cuanto a repuestos, lubricantes, personal especializado, etc., lo que ayuda a minimizar el costo y el impacto sobre la continuidad de las operaciones [2].

### Mantenimiento predictivo

Basado en herramientas como imágenes de rayos x, ultrasonido, análisis de vibraciones y otras, el mantenimiento predictivo diagnostica el desgaste y la

probabilidad de fallo de los diferentes componentes de las máquinas y equipos, en especial, de las partes móviles con mayor riesgo de desgaste. Para realizarlo, se requiere un equipo especializado y personal idóneo que analice la información. Es relativamente costoso, aunque su potencial permite grandes ahorros en operaciones correctivas. La suspensión y reprogramación de actividades productivas son mínimas [3].

### Mantenimiento productivo total (TPM)

Hace parte del conjunto de herramientas de la calidad total de la escuela japonesa. Permea toda la organización con un conjunto de principios como: mantenimiento autónomo, mantenimiento planeado, administración temprana de los equipos nuevos, eliminación de los problemas vitales, educación y entrenamiento en el trabajo. Su filosofía trasciende los departamentos de mantenimiento hasta llegar a la administración, y de esta forma crea en todo el personal una conciencia común, orientada a minimizar los fallos y mantener la planta en funcionamiento [4].

### Programación de actividades

Las técnicas de programación de actividades permiten generar esquemas de asignación de tareas y secuenciación de operaciones que posibilitan un eficiente uso de los recursos y se logra cumplir con las actividades en los tiempos requeridos. Aunque existen técnicas de programación que usan métodos exactos, la mayoría de los problemas de programación son de carácter no polinomial (NP-Hard), por lo que su complejidad crece exponencialmente al crecer su tamaño, de modo que se requieren técnicas inteligentes como las metaheurísticas para su solución [5].

El presente trabajo es una propuesta de aplicación de una técnica avanzada de solución a problemas de optimización, en función del mejoramiento de la programación de actividades de mantenimiento preventivo, con potencial de aplicación en diferentes empresas industriales y comerciales.

Se presenta una técnica avanzada para programar las actividades de mantenimiento, de forma que se minimiza el máximo tiempo requerido para mantenimientos preventivos por semana de trabajo, lo que redunda en un flujo de programación del mantenimiento más eficiente, sin picos de alta carga de trabajo para los recursos dedicados al mantenimiento y sin subutilización de los recursos.

## 2. MATERIALES Y MÉTODOS

Se parte de un conjunto de datos de mantenimiento conformado por la lista de equipos, actividades y una tabla de programación del mantenimiento, conformada por las actividades requeridas con su duración estimada y su frecuencia. Se formula el problema como un problema de optimización, en el que las variables de decisión son las semanas iniciales de programación de las actividades. La función objetivo es minimizar el máximo de tiempo total dedicado al mantenimiento programado en una semana. Se hace el cálculo para una cantidad de semanas igual al mínimo común múltiplo de las frecuencias de mantenimiento de todas las actividades a programar en todas las máquinas. De esta manera, se asegura que la función objetivo considerará el peor de los casos de coincidencia de carga de mantenimiento en una misma semana para una programación suficientemente anticipada.

El algoritmo tiene como cromosoma un conjunto de valores de semanas iniciales de programación para cada combinación equipo-actividad. Cada uno de los genes del cromosoma es la semana inicial propuesta para la programación de esa actividad. Los padres se seleccionan por ruleta proporcional al valor de la función objetivo (minimizar el máximo tiempo total semana de mantenimiento). El cruzamiento se hace en varios puntos de cruce por un operador SJ2OX. La mutación usa el operador ARB2, con una probabilidad de mutación de 0,3 para una tasa efectiva de mutación de 0,003 en un caso de 100 actividades a programar. La actualización de la población es por torneo entre cada descendiente y el individuo menos apto de la población previa (mayor tiempo total semanal de

mantenimiento). El criterio de parada es un número determinado de iteraciones, sin mejorar el valor de la función objetivo del mejor individuo de la población. El número de iteraciones se puede determinar a partir de los parámetros del problema. La solución final es una lista de semanas iniciales de programación de las actividades de mantenimiento en las máquinas, que complementa la tabla de programación y permite calcular la programación detallada semana por semana.

En la Tabla 1 se describen las variables y los parámetros utilizados en el modelo matemático del algoritmo.

**Tabla 1.** Variables y parámetros

Variables	Significado
$x_{ij}$	Semana inicial de programación de la actividad $i$ en la máquina $j$
Parámetros	Descripción
$I$	Índice de la actividad
$J$	Índice de la máquina
$K$	Índice de la semana
$d_{ij}$	Duración estimada de la actividad $i$ en la máquina $j$
$f_{ij}$	Frecuencia (en semanas) con la que debe hacerse la actividad $i$ en la máquina $j$
$T_k$	Tiempo total de programación de mantenimiento para la semana $k$

Se establece el problema de optimización como

$$\text{Min } z = \text{Máx} (T_k) \quad (1)$$

Sujeto a

$$T_k = \sum_{i=1}^n \sum_{j=1}^m Si(k = x_{ij} + Wf_{ij}, d_{ij}, 0) \quad (2)$$

$$\begin{aligned} k &= 1, 2, \dots, n \\ W &= \text{número entero mayor que } 0 \\ x_{ij} &\geq 0 \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \end{aligned} \quad (3)$$

Para la solución, se plantea un algoritmo genético estándar. Los algoritmos genéticos funcionan de forma similar a la reproducción de los seres vivos. Se hace una analogía en la que cada uno de los genes de cada cromosoma representa una de las variables de la solución. La función objetivo se evalúa con los valores de las variables correspondientes al individuo.

Se propone un cromosoma que corresponde a un conjunto de semanas iniciales de programación de las actividades de mantenimiento. El primero de los genes es la semana en la que se programará por primera vez esta actividad en la primera máquina, el segundo es la semana para la segunda actividad de esa misma máquina y así sucesivamente, hasta programar la semana inicial de la última actividad de la última máquina. Los valores posibles para los cromosomas son números enteros mayores o iguales que 1 y menores o iguales que la frecuencia de repetición de la actividad

con esa probabilidad se elige como padre. En cada generación se eligen 2 padres para generar dos descendientes.

### Cruzamiento

Siguiendo el operador SJkOX, adaptado de [6], se copian primero en el mismo orden los elementos que están en ambos padres en los mismos lugares. Luego se toma para el primer descendiente la misma secuencia del primer parente en el mismo orden, hasta el punto de cruce. De allí en adelante se complementa con los genes no incluidos hasta el momento, tomándolos del otro parente en su mismo orden relativo. Para el segundo descendiente, se toman del segundo parente en la misma secuencia a partir del punto de cruce y se complementan los genes faltantes, incluyéndolos antes del primer punto de cruce en el mismo orden relativo que tienen en el primer parente.

### Población inicial

Inicialmente, se genera un conjunto de individuos ordenados por criterios empíricos, tales como: frecuencia de programación, duración, código o nombre de la máquina o de la actividad, etc., tanto en forma ascendente como descendente. La población restante se genera usando criterios meramente aleatorios.

### Selección de padres

Se aplica una ruleta proporcional al valor de la función objetivo, en la que tienen mayor probabilidad de ser seleccionados los individuos que cuentan con menor tiempo total de desplazamiento. A cada individuo le corresponde una probabilidad inversamente proporcional a su valor de función objetivo (que se desea minimizar). De esta forma, las mejores soluciones (con menor tiempo total programación de mantenimiento en la semana más cargada) tendrán mayor probabilidad de ser elegidas. La elección se hace generando números aleatorios entre 0 y 1. El valor del número aleatorio se compara con la probabilidad acumulada y el individuo

### Mutación

Con una probabilidad  $p$ , menor que 0.3, se hace mutación intercambiando arbitrariamente de posición dos productos seleccionados aleatoriamente dentro del descendiente, se aplica el operador ARB2 [7].

### Actualización de la población

Ambos descendientes compiten por aparte contra el elemento de mayor valor de la función objetivo de la población previa. El ganador del torneo queda en la población y el otro se desecha.

### Criterio de parada

El proceso se repite hasta que pase un número fijo de iteraciones, sin llegar a un nuevo mejor valor de la función objetivo que supere al mejor conocido hasta ese momento.

### 3. EXPERIMENTACIÓN Y RESULTADOS

El algoritmo se implementó en un aplicativo de software diseñado por los autores y se aplicó a un conjunto de datos basado en una empresa del sector agroindustrial.

El conjunto de datos de entrada se presenta en la Tabla 2 y representa un conjunto de actividades de mantenimiento a programar en un conjunto de máquinas.

**Tabla 2.** Actividades a programar

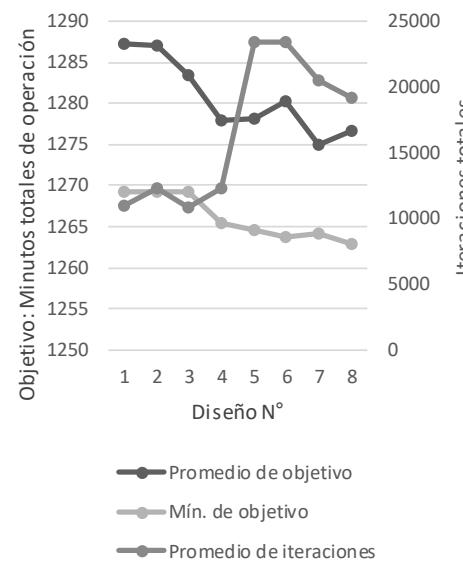
Equipo	Actividad	Duración	Frecuencia
22303-15	L09	1,00	24
22303-15	M07	0,17	24
EE104-99	E08	0,00	156
EE104-99	M07	0,17	156
EE104-99	E05	0,00	156
EE105-99	E08	0,00	52
EE105-99	M07	0,17	52
EE106-99	E08	0,00	52
EE106-99	M07	0,17	52
07307-00	L09	1,00	48
07307-00	M07	0,17	48
07307-10	L09	1,50	48
07307-10	M07	0,17	48
EN162-99	L09	2,00	24
EN162-99	M07	0,17	24
EN162-99	M06	0,08	24
25302-40	L09	2,50	8
25302-40	M07	0,17	8
25301-40	L09	2,50	8
25301-40	M07	0,17	8

El algoritmo se ejecutó con 8 configuraciones diferentes de acuerdo con un diseño  $2^3$ , aplicando el procedimiento mostrado en [8] de la forma que se muestra en la Tabla 3.

**Tabla 3.** Diseño experimental de las pruebas

Diseño	Población (A)	Prob Mutación (B)	Iteraciones (C)
1 (1)	10	0,1	1000
2 a	19	0,1	1000
3 b	10	0,2	1000
4 ab	19	0,2	1000
5 c	10	0,1	2000
6 ac	19	0,1	2000
7 bc	10	0,2	2000
8 abc	19	0,2	2000

Cada una de las combinaciones se probó 30 veces y se obtuvieron los resultados presentados de manera resumida en la Fig. 1.



**Fig. 1:** Resultados en minutos totales de operación e iteraciones totales para llegar al resultado

Se puede apreciar que el algoritmo consigue una respuesta mejor cuando se trabaja con un mayor número de generaciones y una mayor probabilidad de mutación. Según el análisis de varianza que se muestra en la Tabla 4, los factores principales B (la tasa de mutación) y C (las iteraciones) y la interacción AC (población, iteraciones) muestran probabilidades inferiores a  $\alpha$  y diferencias significativas que indican la alta incidencia de esos factores en el resultado. El mayor aporte lo muestran los factores B y C con el 7,8% y el 12,6%

de participación en la suma de cuadrados total. Como en ambos casos los efectos netos en los contrastes son negativos, ambos contribuyen a minimizar el valor de la función objetivo, cuando están en su nivel alto (Probabilidad de mutación = 0,2 y 2000 iteraciones).

**Tabla 4.** Análisis de varianza función objetivo

Factor	Suma Cuadrados	%	gl	Cuadrado Medio	F	Prob
A	15,59	0,1%	1	15,59	0,25	0,62
B	1497,45	7,8%	1	1497,45	23,84	0,00
C	2429,84	12,6%	1	2429,84	38,68	0,00
AB	128,26	0,7%	1	128,26	2,04	0,15
AC	334,53	1,7%	1	334,53	5,33	0,02
BC	149,39	0,8%	1	149,39	2,38	0,12
ABC	82,08	0,4%	1	82,08	1,31	0,25
ERROR	14572,54	75,9%	232	62,81		
TOTAL	19209,68	100,0%	239			

Los resultados corroboran la hipótesis de que es posible mejorar la eficiencia de la programación del mantenimiento. Si se comparan los resultados del algoritmo con otras alternativas como los ordenamientos por variables propias del problema (la frecuencia o la duración), tanto en forma ascendente como descendente, se encuentra un tiempo total de operación entre un 8% y un 20% menor con la aplicación del algoritmo, como se aprecia en la Fig. 2.

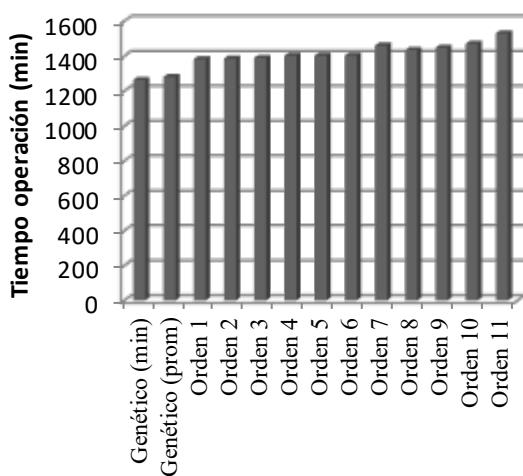


Fig. 2: Tiempos totales de operación según diferentes ordenamientos

Se prueba que se logra reducir la variación de tiempo entre las diferentes semanas de programación de mantenimiento si se tiene una programación equilibrada sin exceso de tiempo ocioso ni saturación de los recursos. Aunque el algoritmo en su diseño tiene componentes aleatorios y, por lo tanto, no garantiza el óptimo global ni permite conocer qué tanto se acerca a este, sí hay consistencia entre los resultados que muestran valores promedio significativamente mejores que las alternativas de ordenamiento contra las que fueron comparadas, y contra el ordenamiento aleatorio que representaría una asignación del espacio sin utilizar ninguna heurística ni herramienta de optimización.

De otro lado, el tiempo de procesamiento de la solución es competitivo en condiciones de entorno real, dado que genera soluciones aplicables en dos minutos o menos. Esto hace que la solución sea adaptable a la gestión de la programación de actividades de mantenimiento y puede proyectarse a otros escenarios.

#### 4. TRABAJOS FUTUROS

Este trabajo ha sido el abre bocas para una segunda fase en la cual se puede extender el proyecto a otras zonas del país, pero también se puede ampliar la aplicación a otros sectores de la industria.

#### 5. CONCLUSIONES

Se formula el problema de programación de mantenimiento como un modelo de optimización; se tiene como función objetivo minimizar el máximo tiempo semanal de mantenimiento preventivo. Dada la naturaleza combinatoria del problema, se propone una metaheurística (algoritmo genético) como estrategia de solución.

Los tiempos de procesamiento son apropiados para el uso de esta herramienta en el entorno real de programación del mantenimiento, dado que los resultados, en promedio, se obtienen en tiempos computacionales entre 20 y 135 segundos, por lo que facilita la repro-

gramación y los ajustes cada que sean necesarios. El resultado del algoritmo muestra que, en promedio, se puede obtener soluciones más equilibradas, entre un 8% y 20% mejores que otras opciones como utilizar heurísticas de ordenamiento o soluciones aleatorias. La reducción de esos tiempos impactará tanto la eficiencia y el costo de operación del mantenimiento como la posibilidad de utilizar mejor los recursos de esa área de la empresa.

## 6. AGRADECIMIENTOS

Los autores reconocen el apoyo de la Universidad Católica de Manizales para el desarrollo de esta investigación.

## 7. CONFLICTO DE INTERESES

Los autores declaran la inexistencia de conflicto de interés con institución o asociación comercial de cualquier índole.

## 8. REFERENCIAS

- [1] M. Fernández Cabanas, M. García Melero, G. A. Orcajo, J. M. Cano Rodríguez, y J. Solares Sariego, *Técnicas para el mantenimiento y diagnóstico de máquinas eléctricas rotativas*. Barcelona: Marcombo, 1998. Disponible en: <https://books.google.com/books?id=X3p4bZfoqgEC&pgis=1>
- [2] F. C. Gómez de León, *Tecnología del mantenimiento industrial*. España: Universidad de Murcia, 1998. Disponible en: <https://books.google.com/books?id=bOrFC3532MEC&pgis=1>
- [3] J. L. Pombo, P. A. Laura, M. J. Maurizi, y L. E. Lui-soni, *Mantenimiento predictivo mediante el análisis de vibraciones de máquinas*. Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, 1978. Disponible en: <https://books.google.com.co/books?id=-HBjAAAAMAAJ>
- [3] C. A. Montilla, *Fundamentos de Mantenimiento Industrial* (1a ed.). Pereira , Editorial Universidad Tecnológica de Pereira, 2016.
- [4] S. Nakajima, *Introducción al TPM (Mantenimiento Productivo Total)*. Tecnologías de Gerencia y Producción, 1992. Disponible en: <https://books.google.com.co/books?id=sfBpSwAACAAJ>
- [5] R. A. Gallego, A. Escobar, y R. Romero, *Técnicas de optimización combinatorial*. Universidad Tecnológica de Pereira, 2006.
- [6] R. Ruiz, R., and C. Maroto, "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility". *European Journal of Operational Research*, vol. 169, no. 3, pp. 781–800, 2006. doi: 10.1016/j.ejor.2004.06.038
- [7] T. Murata, and H. Ishibuchi, "Positive and negative combination effects of crossover and mutation operators in sequencing problems". In *Proceedings of IEEE International Conference on Evolutionary Computation* (pp. 170–175). IEEE. doi: 10.1109/ICEC.1996.542355
- [8] D. C. Montgomery, *Diseño y análisis de experimentos* (2da ed.). México D. F.: Limusa Wiley. Disponible en: <https://juarezrd.files.wordpress.com/2013/09/diseno-de-experimentos-montgomery.pdf>
- [10] Yu, Y., and R. B.M. de Koster, "On the suboptimality of full turnover-based storage", *International Journal of Production Research*, vol. 51, no. 6, pp. 1635-1647. doi: 10.1080/00207543.2011.654012.