



**REPROGRAMACIÓN DE PROYECTOS CON RECURSOS RESTRINGIDOS Y
MÚLTIPLES MODOS**

**Presentado a:
GONZALO MEJÍA DELGADILLO, Ph.D.**

**Por:
JULIO CESAR BERRÍO GARCÍA**

TRABAJO DE TESIS

**UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA INDUSTRIAL
ÁREA DE PRODUCCIÓN Y LOGÍSTICA
BOGOTÁ D.C., ENERO DE 2007**



**REPROGRAMACIÓN DE PROYECTOS CON RECURSOS RESTRINGIDOS Y
MÚLTIPLES MODOS**

JULIO CESAR BERRÍO GARCÍA

**Trabajo de grado para optar al
Título de Magíster en Ingeniería Industrial**

**Asesor:
GONZALO MEJIA DELGADILLO, Ph.D.**

**UNIVERSIDAD DE LOS ANDES
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA INDUSTRIAL
AREA DE PRODUCCIÓN Y LOGÍSTICA
BOGOTÁ D.C., ENERO DE 2007**

TABLA DE CONTENIDO

INTRODUCCIÒN	5
PREGUNTA DE INVESTIGACIÒN	8
OBJETIVO GENERAL	8
OBJETIVO ESPECÌFICOS	8
1. REVISIÒN BIBLIOGRÁFICA	9
2. FORMULACIÒN MATEMÁTICA DEL PROBLEMA MMRC PSP	13
3. ALGORITMO GENÉTICO PARA PROGRAMACIÒN DEL PROBLEMA MMRC PSP	16
3.1 Etapa de preprocesamiento	17
3.2 Representaciòn de la soluciòn	18
3.3 Generaciòn de la poblaciòn inicial	19
3.4 Cálculo de la funciòn objetivo de la poblaciòn	20
3.5 Creaciòn de nuevas generaciones a partir del cruce de cromosomas	22
3.6 Mutaciòn de cromosomas	24
4. REPROGRAMACIÒN DE PROYECTOS BAJO EL MODELO MMRC PSP	26
4.1 Heurística para la reprogramaciòn de proyectos bajo el esquema MMRC PSP	27
4.2 Algoritmo Genético para la reprogramaciòn de proyectos bajo el esquema MMRC PSP	32
4.2.1 Ajuste en el cromosoma	32
4.2.2 Cruce entre cromosomas	33
4.2.3 Ajuste en la mutaciòn de cromosomas	33
4.2.4 Procedimiento de reemplazo de modos	33
5. RESULTADOS COMPUTACIONALES	34
5.1 Configuraciòn computacionales del Algoritmo Genético para la programaciòn de proyectos	34
5.2 Resultados computacionales del Algoritmo Genético para la programaciòn de proyectos	38
5.3 Comparaciòn resultados reprogramaciòn de proyectos utilizando un Algoritmo Genético versus una técnica heurística	39
6. CONCLUSIONES	49
7. BIBLIOGRAFÍA	51

TABLA DE ANEXOS

ANEXO 1. GRÁFICOS INTERVALOS DE MEDIAS DE LOS TRATAMIENTOS PARA LOS DIFERENTES TAMAÑOS DE PROYECTOS	53
ANEXO 2. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 16 ACTIVIDADES CON ALGORITMO GENÉTICO	54
ANEXO 3. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 20 ACTIVIDADES CON ALGORITMO GENÉTICO	55
ANEXO 4. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 30 ACTIVIDADES CON ALGORITMO GENÉTICO	56
ANEXO 5. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 16 ACTIVIDADES	57
ANEXO 6. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 20 ACTIVIDADES	58
ANEXO 7. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 30 ACTIVIDADES	59

INTRODUCCIÓN

El logro de los objetivos y metas definidas por la alta dirección de cualquier organización, ha influenciado el cambio en la manera de entender las relaciones entre las diferentes dependencias de las compañías. Esto ha dado como resultado la estructuración de las actividades de la organización a través de proyectos con resultados deseados específicos acordes con los objetivos arriba mencionados.

Como lo expresa Hartmann (1999) los proyectos pueden encontrarse en varias áreas como la construcción de edificios, puentes y autopistas, en la planeación de la producción, en desarrollos e investigaciones científicas, en el montaje de eventos deportivos, etc.

El desarrollo de proyectos es un proceso de concepción, preparación, organización, manejo y control de las transformaciones necesarias de un sistema para alcanzar un objetivo (Tavares, 20002). Aunque no está mencionado en el concepto, la programación de los proyectos a realizar también es una tarea central del administrador de proyectos. El objetivo de la programación es definir un plan periódico que permita identificar las fechas de inicio y finalización de los proyectos teniendo en cuenta restricciones de recursos y tratando de minimizar la fecha de terminación de los proyectos, retardos, tardanzas, etc.

En Chase (2001) se expresa que inicialmente las técnicas que se utilizaron para planear y controlar proyectos fueron el PERT y el CPM. La técnica PERT (Program Evaluation and Review Technique o Técnica para la Revisión y Evaluación de Programas) era utilizada en la programación de proyectos avanzados bajo incertidumbre, teniendo como principal característica que se usaban tres estimaciones (optimista, pesimista y mejor) para medir el tiempo de una actividad. Por otra parte, según el autor, la técnica CPM (Critical Path Method o Método de la Ruta Crítica) servía para programar actividades rutinarias de plantas y tenía como característica que para determinar el tiempo de ejecución de una actividad se usaba la mejor estimación.

A medida que la complejidad en la gestión y control de las actividades de los proyectos fue aumentando, se hizo necesario que matemáticos e ingenieros, apoyados en herramientas de programación de computadores iniciaron el modelamiento matemático de problemas involucrados con la programación de proyectos. Alcaraz et al (2003) mencionan que la versión clásica del problema de programación de proyectos es el Resource Constrained Project Scheduling Problem (RCPS) o problema de programación de proyectos con recursos restringidos.

En el problema RCPS se tiene un conjunto de actividades las cuales están relacionadas entre sí por restricciones de precedencia. Igualmente para la ejecución de las actividades del proyecto es necesario el consumo de recursos los cuales tienen disponibilidad restringida. La función objetivo del problema es minimizar el makespan o el mayor tiempo de finalización de las actividades.

Una versión ampliada del problema RCPSP es el Multi Mode Resource Constrained Project Scheduling Problem (MMRCPSP) o Problema de Programación de Proyectos con Recursos Restringidos y múltiples modos. Como en el RCPSP, en este problema las actividades están relacionadas entre sí por restricciones de precedencia. Por otra parte en el problema MMRCPSP las actividades del proyecto se pueden ejecutar en diferentes modos. Los modos representan las diferentes maneras en que una misma actividad puede ser ejecutada. Cada posible modo de ejecución de una actividad contiene información sobre la duración y consumo de recursos durante la realización de esta. La función objetivo del problema MMRCPSP es minimizar el makespan del proyecto.

Aunque la programación de los proyectos es un paso importante para la gestión de las actividades de la empresa, es importante tener en cuenta que el cambio es la constante por lo cual la reprogramación de tareas en cualquier instante del tiempo se ha convertido en un objetivo claro para los administradores de proyectos.

Abumaizar et al (1997) menciona que en ambientes dinámicos como los Job Shops (problema que consiste en programar n tareas de diferentes trabajos en m máquinas; las tareas no pueden ser procesadas dos veces por la misma máquina), tan pronto los programas de producción son finalizados, estos se ven sujetos a cambios aleatorios lo cual implica que el programa original se vuelve obsoleto por lo que toma importancia el hecho de estudiar maneras de reprogramar las tareas teniendo como objetivo hacer el menor número de cambios posibles en las fechas de inicio y finalización de estas.

Esta misma casuística puede ocurrir cuando se están programando proyectos. Es claro, que hay incertidumbre sobre el desarrollo de las actividades de un proyecto puesto que no se garantiza que las condiciones económicas, políticas y sociales del medio permanezcan sin cambios. Por tal razón en los últimos años autores como Calhoun et al (2002) han trabajado en diseñar algoritmos computacionales que permitan reprogramar el plan de ejecución de las actividades de un proyecto cuando este se ve interrumpido por eventos no esperados.

En esta investigación se explora el uso de la metaheurística Algoritmo Genético con el objetivo de reprogramar el plan de ejecución de un proyecto el cual se ve interrumpido por la llegada de una nueva actividad la cual tiene prioridad de ejecución. También se plantea una heurística de reprogramación basada en el trabajo de Abumaizar et al (1997) la cual solo reprograma aquellas actividades que son afectadas directamente o indirectamente por la interrupción. Con el fin de generar soluciones iniciales para resolver el problema de reprogramación se plantea un Algoritmo Genético competitivo que resuelva el problema de programación inicial.

Se plantea un diseño de experimentos con el fin de comparar ambos algoritmos de reprogramación. Se utilizan indicadores de cambio en la función objetivo y en la estabilidad del programa como medidas de desempeño de las técnicas propuestas.

El presente documento esta distribuido de la siguiente forma: inicialmente se planteará la pregunta y los objetivos de la investigación. En el numeral 1 se documenta los principales aportes realizados por diferentes investigadores que alimentaron la investigación. En numeral 2 se explica el funcionamiento del Algoritmo Genético que programará proyectos bajo el esquema MMRCPS. En el numeral 3 se documenta el Algoritmo Genético y la heurística propuesta para la reprogramación de proyectos. En el numeral 5 se muestran los resultados computacionales de la investigación y las comparaciones de las heurísticas de reprogramación. Finalmente se resuelve la pregunta de investigación y plantean las conclusiones del estudio en el numeral 6.

PREGUNTA DE INVESTIGACIÓN

Qué tan bueno son los resultados de reprogramar un proyecto bajo el esquema MMRCPS utilizando un Algoritmo Genético si se comparan con los resultados de una técnica heurística que reprograma solamente aquellas actividades afectadas por la interrupción?

OBJETIVO GENERAL

Diseñar la reprogramación de un proyecto enmarcado bajo el problema MMRCPS utilizando Algoritmos Genéticos versus una técnica heurística que reprograma aquellas actividades afectadas por la interrupción.

OBJETIVOS ESPECÍFICOS

Proponer e Implementar un Algoritmo Genético que permita reprogramar proyectos enmarcados en el problema MMRCPS.

Proponer e implementar una técnica heurística que permita reprogramar proyectos a partir de un programa inicial establecido.

Diseñar indicadores que permitan medir la efectividad de los algoritmos planteados para reprogramar proyectos enmarcados en el problema MMRCPS.

1. REVISIÓN BIBLIOGRÁFICA

Con el fin de resolver el problema de programación de proyectos con recursos restringidos y múltiples modos los investigadores han aplicado métodos exactos y métodos heurísticos.

Alcaraz et al (2003) menciona que son pocos los métodos exactos que se han implementado para resolver el problema MMRCPS. Según el mismo autor el algoritmo Branch-and-Bound con límite de tiempo propuesto por Sprecher y Drexler es el método exacto más efectivo que se ha diseñado puesto que su comportamiento es cercano al de una heurística.

La poca atracción de los investigadores por el uso de métodos exactos para resolver el problema MMRCPS se puede deber a que ha sido difícil encontrar óptimos para problemas altamente restringidos con más de 20 actividades y tres modos por actividad, como lo expresa Hartmann (2001).

Ya Blazewicz (1983) había demostrado que el problema RCP es un problema NP-Hard. Un problema se denomina NP-Hard si no existe un algoritmo conocido que obtenga un programa óptimo para cualquier instancia de este en tiempo polinomial. Como lo expresa Hartmann (1999) es claro que en el caso del MMRCPS, si hay un modo por actividad y no existen recursos no renovables se obtiene el problema RCP. Por lo tanto el problema MMRCPS es también NP-Hard.

El hecho de que el problema MMRCPS sea NP-Hard ha conducido a que los investigadores hayan preferido el uso de técnicas heurísticas para obtener programas cercanos al óptimo para proyectos de gran tamaño. Entre las heurísticas que se han utilizado para resolver el problema MMRCPS encontramos enfriamiento simulado, procedimientos de búsqueda local, heurísticas que resuelven el problema sin tener en cuenta recursos no renovables, etc (Hartmann, 1999).

Igual que otros sistemas de producción, la programación de proyectos también se ve afectada por el ambiente dinámico en el cual se ejecutan las diferentes actividades. Esto conlleva a que sea necesaria la actualización del cronograma del proyecto lo cual se denomina reprogramación.

Vieira et al (2003) recopilaron en su artículo las principales causas por las cuales se debe reprogramar el cronograma de operación de un sistema de producción general entre las cuales encontramos:

- Falta de maquinaria
- Llegada de tareas urgentes
- Cancelación de tareas
- Cambio en las fechas definidas para la finalización de tareas (adelanto o retraso)
- Retraso en la llegada de la materia prima
- Reprocesos por problemas de calidad
- Sobre o subestimación de los tiempos de ejecución de las actividades

- Ausencia de operarios

De igual manera Vieira et al (2003) hacen un resumen de los principales conceptos concernientes al tema de la reprogramación entre los que se encuentran:

Ambiente de reprogramación: Identifica los trabajos que deben ser reprogramados. Dependiendo del sistema de producción puede haber finitos o infinitos trabajos a reprogramar.

Estrategias de reprogramación: Hay dos estrategias de reprogramación. La programación dinámica cuyo objetivo es programar el trabajo en el preciso instante en que sea necesario utilizando información disponible en el momento lo que implica no hay un programa original. Por otra parte se encuentra la estrategia de reprogramación reactiva en la cual primero se programan los trabajos y luego generada la interrupción se utilizan diferentes métodos para realizar la reprogramación.

Métodos de reprogramación: En general hay tres métodos de reprogramación. El primer método denominado Right-Shift Rescheduling retrasa el cronograma de producción el tiempo que dure la interrupción. El Partial Rescheduling solo reprograma aquellas tareas que se ven afectadas directa o indirectamente por la interrupción. El Complete Regeneration reprograma totalmente las tareas a partir del momento en que se genera la interrupción.

Punto de programación: Instante en el tiempo que el programa es creado o revisado.

Periodo de reprogramación: Intervalo de tiempos entre dos puntos de programación.

Estabilidad del programa: Medición del número de cambios que sufre un programa cuando es expuesto a revisiones.

Robustez del programa: Medición de la cantidad de interrupciones que serian necesarias para degradar el programa original.

En cuanto a investigaciones, desde principios de los años 90 diferentes autores han abordado el tema de la reprogramación. Kwei Li te al (1993) desarrollaron un algoritmo adaptado de los sistemas MRP con el fin de reprogramar solamente aquellas tareas que fueron impactadas por el cambio en el programa original. Los autores desarrollaron este algoritmo con el fin de ofrecer alternativas al uso de métodos manuales y computacionales para realizar la reprogramación. En este paper se explica que los sistemas manuales son tediosos e ineficientes debido a que una persona con cierta experiencia debe editar la programación original solamente basado en un gráfico. Los métodos computacionales como el diagrama Gantt electrónico y las aproximaciones por simulación aun que reducen las manualidades, dejan en el programador la tarea de identificar las actividades impactadas.

Jain et al (1997) proponen la utilización de algoritmos genéticos con el fin de reprogramar la producción cuando se presentan situaciones especiales como salida de servicio de máquinas, cambio en prioridades de trabajos, llegadas de nuevos trabajos y cancelaciones de orden.

Abumaizer et al (1997) proponen un algoritmo para reprogramar operaciones en un Job Shop. Este algoritmo solo reprograma aquellas actividades que sean afectadas directa o indirectamente por la interrupción. Este algoritmo se compara con otro que reprograma todas las actividades sin importar si se ven afectadas y con un algoritmo que retrasa las actividades no iniciadas el tiempo de duración de la interrupción. Con el fin de determinar cual de los algoritmos tiene mejor desempeño se plantea un diseño de experimentos en el cual se usan como medida indicadores de eficiencia y estabilidad del proyecto reprogramado. El resultado de esta investigación indica que el algoritmo que reprograma solamente las actividades afectadas es el que mejor desempeño presenta.

Bierwirth et al (1999) proponen la implementación de un Algoritmo Genético para reprogramar Job Shops en ambientes no determinísticos. Los autores plantean que la llegada de trabajos al piso de producción es imprevista por lo que los requerimientos específicos para su procesamiento no son conocidos con anterioridad. Los resultados de la investigación indican que los métodos tradicionales de control de la producción son superados por el Algoritmo Genético.

Wong et al (2004) desarrollaron un algoritmo genético para reprogramar operaciones mediante segmentación en tiempo real. Esta investigación concluyó que el algoritmo genético puede reducir el makespan (máximo tiempo de terminación de los trabajos programados) y la influencia causada por el cambio en los programas.

Mason et al (2004) mencionan que en ambientes dinámicos, una vez el programa es lanzado al piso de trabajo, está inmediatamente sujeto a interrupciones que pueden acusar que el programa inicial se vuelva obsoleto. Para el caso especial de los Job Shops complejos los autores implementan tres algoritmos de reprogramación basados en el trabajo de Abumaizar et al (1997). El resultado indica que el algoritmo que reprograma todas las actividades entrega mejores resultados en comparación con los demás aunque el consumo de tiempo es mucho mayor.

Aunque la literatura del problema de reprogramación de operaciones es amplia, infortunadamente poca de esta enfrenta el problema de replanear o reprogramar proyectos.

Calhoun et al (2002) proponen el uso de la metaheurística búsqueda tabú junto a programación por objetivos para reprogramar proyectos en mercados en el problema MMRCPS. En esta investigación se denomina reprogramación a la reasignación de actividades de un proyecto en circunstancias de "peligro" durante la ejecución del programa original.

Demeulemeester et al (2002) y Alvarez (2004) concluyeron que las técnicas que mas se utilizan para la programación y reprogramación de proyectos son las reglas de despacho y metaheurísticas como enfriamiento simulado y algoritmos genéticos

2. FORMULACIÓN MATEMÁTICA DEL PROBLEMA MMRCPSP

Como se expresó con anterioridad el problema base de esta investigación es el Multi Mode Resource Constrained Project Scheduling Problem o Problema de programación de proyectos con recursos restringidos y múltiples modos. En resumen el problema consiste en minimizar el máximo tiempo de finalización de las actividades de un proyecto las cuales se encuentran restringidas por relaciones de precedencia (una actividad no puede ser programada hasta que todos sus predecesores lo hayan sido) y por la disponibilidad de recursos renovables, no renovables y doblemente restringidos.

Para que una actividad se pueda llevar a cabo deben consumirse los tipos de recursos antes mencionados. A continuación se detalla cada uno de los tipos de recursos.

- Recursos renovables: su cantidad está limitada en ciertos períodos de tiempo. Por ejemplo para un proyecto de construcción se tiene disponibilidad de 100 horas-hombre diariamente para todas las actividades programadas. Al comenzar un nuevo día se vuelven a tener disponibles las mismas 100 horas-hombre.
- Recursos no renovables: su cantidad está limitada para el total del proyecto. Siguiendo el ejemplo del proyecto de construcción, los ingenieros tiene disponibles US\$1'000.000 de presupuesto para los gastos generados durante la ejecución de las diferentes actividades. Si antes de finalizar el proyecto se ha consumido el presupuesto este deberá detenerse hasta que se obtengan nuevos recursos.
- Recursos doblemente restringidos: son aquellos que tienen características de recursos renovables y de no renovables. Siguiendo con el ejemplo del dinero para el total del proyecto se tiene un presupuesto de US\$1'000.000 lo cual equivale a un recurso no renovable pero también hay una restricción de que el máximo gasto diario es de \$25.000 lo cual equivale a una restricción de recurso renovable.

La definición de la duración de la actividad y de la cantidad de recursos renovables y no renovables que se consumirían está dado por el modo en que esta se ejecute. Un modo representa las diferentes maneras de realizar una actividad del proyecto. En el proyecto de construcción, por ejemplo, existe una actividad denominada pintura. Hay tres empresas disponibles que podrían realizar esta actividad y que equivaldrían a los modos en que se ejecutaría la actividad pintura. La empresa A envía un presupuesto de gasto de US\$4.000 diariamente y consumo de 40 horas hombre. La empresa B envía un presupuesto de gasto de US\$5.000 y un consumo de 30 horas hombre. La empresa C presupuesta un gasto de US\$7.000 y un consumo de 20 horas hombre. En este caso especial, el objetivo es escoger el modo en que se debe ejecutar la actividad pintado teniendo en cuenta que esta debe iniciar después de la finalización de actividades precedentes (levantamiento de muros, empañotamiento, etc) y cumpliendo la restricciones de disponibilidad de recursos renovables y no renovables lo cual permita que su tiempo de finalización sea el menor posible.

A continuación se presenta un modelo matemático del problema MMRCPSP adaptado del modelo planteado por Hartmann (1999) para el problema MMRCPSP con ventanas de tiempo:

Índices

j : Actividades a programar
 k : Recursos renovables disponibles
 l : Recursos no renovables disponibles
 t : Períodos de tiempo $\{1, \dots, g\}$ donde g es la sumatoria de los mayores tiempos de ejecución de las actividades.

Parámetros

M_j : Modos en los cuales la actividad j puede ejecutarse
 $P_{j,m}$: Tiempo de procesamiento de la actividad j siendo ejecutada en el modo m
 $R_{k,t}$: Cantidad de recursos renovable k disponible en el periodo t
 RNR_l : Cantidad de recursos no renovable l disponible para todo el proyecto
 $r_{j,m,k}$: Cantidad de recurso renovable k consumido por periodo por la actividad j si se ejecuta en el modo m
 $nr_{j,m,l}$: Cantidad de recurso no renovable l consumido por la actividad j si se ejecuta en el modo m .
 S_j : Conjunto de actividades sucesoras de j

Variables

$X_{j,m} = \begin{cases} 1 & \text{si la actividad } j \text{ es ejecutada en el modo } m; \\ 0 & \text{de lo contrario} \end{cases}$

TI_j : Tiempo de inicio de la actividad j

Modelo

F.O

$$\text{Min}[TI_j \cdot X_{j,1}] \quad (1)$$

s.a

$$\sum_{m=1}^{M(j)} X_{j,m} = 1 \quad \forall_j \quad (2)$$

$$(TI_j + P_{j,m}) \cdot X_{j+1,m} < TI_{j+1} \cdot X_{j+1,m} \quad \forall_j \quad \forall_{j+1} \in S_j \quad (3)$$

$$\sum_{j=1}^J \sum_{m=1}^{M(j)} nr_{j,m,l} \cdot X_{j,m} \leq RNR_l \quad \forall_l \quad (4)$$

$$\sum_{j/TI(j) \leq t \leq TI(j)+P(j)}^J \sum_{m=1}^{M(j)} r_{j,m,k} \cdot X_{j,m} \leq R_{k,t} \quad \forall_k \quad \forall_t \quad (5)$$

$$X_{j,m} \in \{0,1\} \quad \forall_j \quad \forall_m \quad (6)$$

$$TI_j \geq 0 \quad \forall_j \quad (7)$$

- (1) La función objetivo se define como la minimización del tiempo de finalización del trabajo ficticio final J . Este trabajo para fines de programa se ejecuta en el modo 1 aunque esto no implique que su duración sea mayor a cero o consume recursos renovables y no renovables
- (2) Restricción que garantiza que cada actividad debe ser ejecutada en un solo modo.
- (3) Restricción que garantiza que el tiempo de inicio de una actividad debe ser mayor que el mayor tiempo de finalización de sus predecesoras por restricción.
- (4) Restricción que garantiza que el consumo de recursos no renovables por parte de las actividades programadas sea menor o igual que la disponibilidad de estos.
- (5) Restricción que garantiza que el consumo de recursos renovables en cada período de tiempo por las actividades ejecutadas en este sea menor o igual a la cantidad de recursos renovables disponibles
- (6) Se define que la variable es binaria
- (7) Se define que la variable es entera positiva

Hay que aclarar que no se tiene en cuenta los recursos doblemente restringidos dentro del modelo ya que estos se pueden trabajar, como ya se mencionó, como recursos renovables y recursos no renovables.

3. ALGORITMO GENÉTICO PARA PROGRAMACIÓN DEL PROBLEMA MMRCPS

Teniendo en cuenta que el problema MMRCPS está catalogado como NP-hard, en esta investigación se propone la utilización de un método computacional basado en la evolución humana llamado Algoritmo Genético para su solución. Tavarez (2002) menciona que técnicas computacionales como Búsqueda Tabú, Enfriamiento Simulado y Algoritmos Genéticos son las más utilizadas para resolver problemas de programación de proyectos ya que tienden a ser rápidas aunque no garantizan la obtención de la solución óptima.

En general, el Algoritmo Genético es un procedimiento de búsqueda estocástica de soluciones que está diseñado para resolver problemas de optimización combinatoria usando el concepto de computación evolutiva, el cual se basa en el proceso de reproducción de especies animales (Zhou et al, 2002).

El funcionamiento de un Algoritmo Genético se basa en la codificación de una solución a un problema de optimización llamada Cromosoma. Las características especiales de cada solución al problema están representadas en cada posición de este que comúnmente se denomina gen. A cada cromosoma le equivale un valor o número correspondiente a la función objetivo del problema que se desea resolver.

El Algoritmo Genético comienza al formarse un grupo de cromosomas o población inicial de soluciones al problema. Dichas soluciones pueden ser creadas aleatoriamente o mediante técnicas heurísticas como reglas de despacho, etc. A partir de esta población inicial se van generando sucesivamente nuevas generaciones de cromosomas a partir del cruce de información de estos (intercambio de genes que producen cromosomas hijos a partir de cromosomas padres). Con el fin de evitar que el algoritmo converja a óptimos locales se mutan a partir de una probabilidad previamente definida los cromosomas mediante el intercambio de posición entre sus genes.

El Algoritmo Genético puede detenerse cuando alcance el número de generaciones predefinido, cuando el tiempo de ejecución sobrepase el estipulado, no se obtenga mejoría en la solución encontrada, etc.

El diseño de un Algoritmo Genético consiste en la definición de aspectos tales como: representación de la solución (cromosoma), generación solución inicial, estrategia de cruce, estrategia de mutación, criterio de finalización, etc. A continuación se presenta el esquema de funcionamiento del Algoritmo Genético propuesto en esta investigación y se detalla cada uno de los pasos a ejecutar:

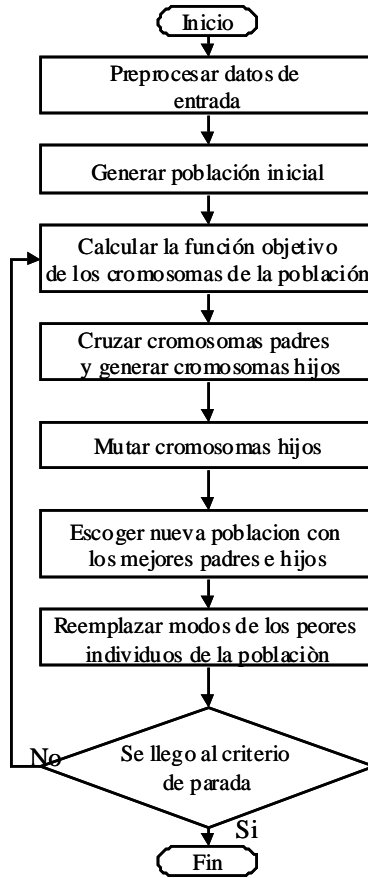


Figura 1. Esquema de funcionamiento del Algoritmo Genético

3.1 Etapa de preprocesamiento

Antes de iniciar la ejecución del Algoritmo Genético se realiza un proceso de depuración de los datos de entrada al algoritmo concerniente a los modos de ejecución de cada una de las actividades del proyecto y sus respectiva duración y consumo de recursos renovables y no renovables.

Este proceso es utilizado por Hartmann (2001) y Alcaraz et al (2003) con el fin de reducir el espacio de búsqueda del algoritmo tratando de eliminar los modos de aquellas actividades que no contiene información con el suficiente valor para restringir las soluciones del problema o para mejorar estas.

Teniendo en cuenta lo estipulado por Sprecher et al (1997) la reducción del espacio de búsqueda se realiza desarrollando los siguientes pasos en orden:

- Eliminando modos no ejecutables. Un modo de una actividad se denomina no ejecutable si al programar esta teniendo en cuenta la información del modo se violan restricciones de consumo de recursos renovables y no renovables.

- Eliminando recursos no renovables redundantes. Un recurso no renovable se denomina redundante si al sumar la máxima utilización de este por cada una de las actividades no se supera la disponibilidad del recurso redundante en cuestión.
- Eliminando modos ineficientes de una actividad. Un modo de una actividad se denomina ineficiente si su duración y consumo de recursos renovables y no renovables son mayores que la de los otros modos de la actividad.

Sprecher et al (1997) indica que el orden de realización de esta depuración debe ser el planteado ya que de lo contrario se podrían presentar interacciones entre la eliminación de modos y recursos no renovables. Por ejemplo la eliminación de un recurso no renovable podría llevar a la ineficiencia de un modo de una actividad.

Dado que Hartmann (2001) menciona que utilizando este preprocesamiento se elimina el 29% de los modos y el bajo esfuerzo computacional necesario para su ejecución se decidió incluirlo en el Algoritmo Genético propuesto.

3.2 Representación de la solución

Como lo afirma Alcaraz et al (2003) en los algoritmos genéticos y en general en todas las metaheurísticas la forma en la cual la solución es representada es crucial para el desempeño del algoritmo.

Como se puede ver en la figura 2, el cromosoma diseñado para este Algoritmo Genético consta de $2N$ posiciones en donde N corresponde al número de actividades del proyecto a programar que incluye dos actividades ficticias que representan el inicio y fin del proyecto.

Las N primeras posiciones del cromosoma contienen la información sobre la prioridad de programación de cada una de las actividades. Las posiciones entre $N+1$ y $2N$ contienen la información de los modos en que serán ejecutados cada actividad.

Por ejemplo la posición $N+2$ del cromosoma contiene el modo en que será ejecutada la actividad B.

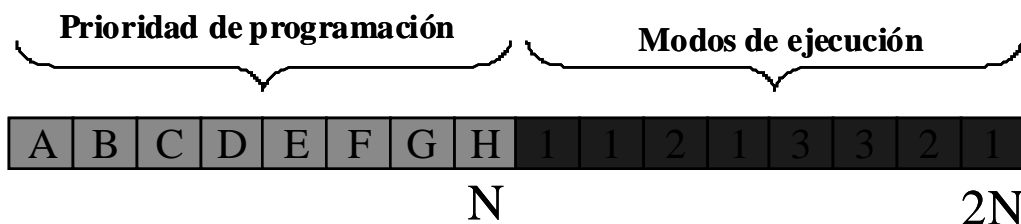


Figura 2. Representación de la solución para el Algoritmo Genético

Es importante aclarar que aunque en el cromosoma se determine que una actividad tiene prioridad de programación sobre otra esto no quiere decir que al momento de la programación del proyecto la actividad con mayor prioridad

inicie su ejecución primero. A continuación se presenta un ejemplo para aclarar esta afirmación en el cual se programarán las actividades B, C, D y E. La información sobre consumo de recursos, duración y restricciones de precedencia de cada una de estas actividades se presenta en la tabla 1.

Actividad	Duración	Consumo de recursos	Sucesoras
B	4	3	--
C	2	3	E
D	2	5	--
E	2	2	--

Tabla 1. Información de entrada para programación de actividades

Siguiendo el orden de programación estipulado en el cromosoma de la figura 2 y la información contenida en la tabla 1, se han programado las actividades B, C, D y E. En la figura 3 se puede observar el diagrama Gantt correspondiente a la programación de las actividades antes mencionadas en donde el eje t corresponde a la duración de las actividades y el eje r corresponde al consumo del recurso renovable

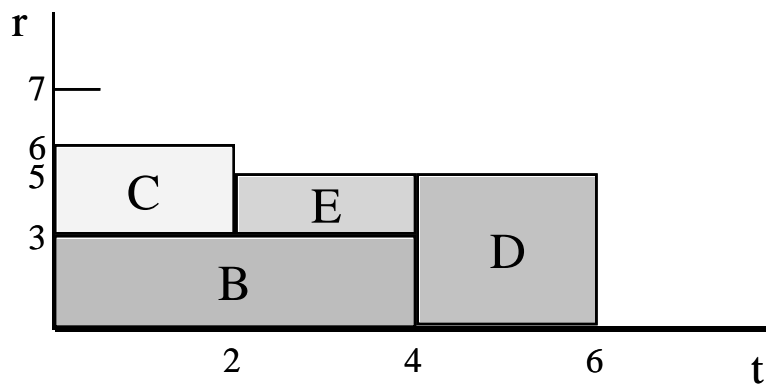


Figura 3. Diagrama Gantt de la programación de las actividades

Como se puede ver en la siguiente figura, aunque la actividad D está delante de la actividad E en el orden de prioridad para programar definido en el cromosoma de la figura 2, la actividad D tiene un tiempo de inicio mayor debido a que solo a partir del instante $t = 4$ se cuenta con los recursos disponibles para su ejecución.

3.3 Generación de la población inicial

El esquema de generación de la población inicial planteado permitirá obtener soluciones infactibles en cuanto al consumo de recursos no renovables con el fin de estudiar un mayor cantidad de individuos diversos que de una u otra manera puedan pasar buena información a individuos factibles.

A diferencia de los trabajos de Alcaraz et al (2003) y Hartmann (2001) en donde primero se definen los modos en que serán ejecutadas cada una de las actividades y luego estas se van programando escogiéndolas de una lista de actividades factibles (aquellas cuyos sucesores ya fueron programados) mediante la regla de LFT (latest finish time), en el Algoritmo Genético propuesto primero

se define el orden en que serán programadas las actividades y luego el modo en que serán ejecutadas cada una de ellas.

Con el fin de generar inicialmente el orden en que serán programadas cada una de las actividades se crea una lista de factibilidad en donde estarán ubicadas aquellas actividades cuyos predecesores ya fueron ingresados en el cromosoma. Aleatoriamente se escoge una actividad de la lista de factibilidad con el fin de que sea ubicada dentro del cromosoma. A continuación se estudian aquellas actividades que no están ni en el cromosoma ni en la lista de factibilidad con el fin de determinar si ya todos sus predecesores fueron programados y así incluirlas dentro de la lista. Estos pasos se realizan hasta que la última actividad se encuentre ubicada en el cromosoma y por lo tanto tenga un orden de programación.

Para iniciar la definición de los modos en que serán ejecutadas cada una de las actividades se escoge aleatoriamente el modo en que se ejecutará la primera actividad a programar después de la actividad ficticia de inicio. A continuación se escoge el modo de la próxima actividad a programar que consuma un menor porcentaje del recurso renovable que en porcentaje mayor consuma la actividad previa. Por ejemplo si la actividad F ubicada en la sexta posición del cromosoma de la figura 2 consume un mayor porcentaje del recurso renovable 1 que del 2, y la actividad G consume el 40%, 55% y 20% del recurso renovable 1 respectivamente en los modos en que se puede programar, se escogerá el último modo para ejecutar la actividad G.

Estos pasos para determinar la población inicial del algoritmo garantizan que se cumplan las restricciones de recursos renovables ya que al momento de decodificar el cromosoma, los tiempos de inicio de las actividades a programar se verán restringidos por los tiempos de finalización de las actividades predecesoras y por la disponibilidad de recursos renovables. En la medida que estas restricciones no se cumplan el inicio de la actividad se retrasará hasta que estas se satisfagan.

Al contrario de lo que ocurre con las restricciones de consumo de recursos renovables el procedimiento de generación de la población inicial de cromosomas no garantiza que se cumpla la restricción de recursos no renovables. Con el fin de cumplir esta restricción se escoge aleatoriamente una actividad del proyecto a la cual se le cambia de modo. Si después de cinco intentos este procedimiento no permite que la solución sea factible en cuanto al consumo de recursos no renovables la solución se acepta y se procede al cálculo de su función objetivo el cual se explicará más adelante.

3.4 Cálculo de la función objetivo de la población

Como se expresó con anterioridad el objetivo de la investigación es programar las actividades de un proyecto las cuales están restringidas por la disponibilidad de recursos renovables y no renovables de tal manera que se minimice el tiempo de finalización del proyecto o lo que es lo mismo que se minimice el tiempo de finalización de la última actividad programada.

Con base en el orden establecido en el cromosoma para la escogencia de una actividad con el fin de programarla y el modo en que se debe ejecutar se van ingresando cada una de las actividades en el programa de ejecución del proyecto. El tiempo de inicio de la actividad inicialmente es cero pero se va atrasando o moviendo hacia la derecha hasta que desde el inicio de la actividad y durante su duración se cumplan las restricciones de secuencia (una actividad debe iniciar después del mayor tiempo de finalización de sus predecesores) y restricciones de recursos renovables (en cada instante de tiempo el consumo de recursos renovables por parte de las actividades en este programadas debe ser menor o igual que la disponibilidad de los recursos).

Como se mencionó anteriormente no se puede garantizar que los cromosomas que se obtengan en la población inicial o en las poblaciones futuras cumplan con las restricciones de consumo de recursos no renovables. Por tal razón se ha definido que estos cromosomas aunque infactibles permanezcan en el algoritmo con el fin de que ofrezcan mayor diversidad de soluciones. Alcaraz et al (2003) y Hartmann (2001) proponen algoritmos que aunque permiten que los cromosomas infactibles no sean eliminados, castigan sus funciones objetivo.

Hartmann (2001) propone que la función objetivo de los cromosomas factibles sea el C_{max} (tiempo de finalización de la actividad ficticia final) y que la función objetivo de los cromosomas infactibles sea la sumatoria de la máxima duración de las actividades del proyecto mas el exceso de recursos no renovables consumidos por el programa planteado en el cromosoma.

Alcaraz et al (2003) propone que la función objetivo de los cromosomas factibles sea el C_{max} mientras que la función objetivo de los cromosomas infactibles sea el máximo C_{max} de los cromosomas factibles de la población más la diferencia entre el C_{max} del cromosomas y la ruta crítica del proyecto mas el exceso de recursos no renovables consumidos por el programa planteado en el cromosoma.

Los autores demuestran que esta propuesta es mucho mas flexible que la de Hartmann (2001) ya que permite que aunque un cromosoma sea no factible, su función objetivo no sea tan grande en comparación con la de los cromosomas factibles con el fin de definir los cromosomas padre y madre que se cruzaran para formar los cromosomas hijos.

En el Algoritmo Genético propuesto a diferencia de los anteriores se trata de reparar el cromosoma un máximo de 5 veces utilizando el mismo procedimiento de reparación de la población inicial. Si finalizado el procedimiento de reparación el cromosoma sigue siendo infactible se procede a castigar su función objetivo. La función objetivo de dicho cromosoma infactible se calcula sumándole al máximo C_{max} de los cromosomas factibles de la población la diferencia entre el C_{max} del cromosoma que se esta evaluando y el mínimo C_{max} de los cromosomas factibles mas el exceso de recursos no renovables consumidos por el programa planteado en el cromosoma.

Las tres propuestas garantizan que la función objetivo de los cromosomas infactibles sea mayor que la máxima función objetivo de los cromosomas

factibles de la población. La diferencia entre la propuesta de esta investigación y las anteriores es que esta es mucho más flexible ya que en las primeras generaciones se escogen para cruzar con mayor probabilidad soluciones malas con el fin de que el algoritmo tenga más opciones de encontrar el óptimo pero a medida que va avanzando, las soluciones malas se van descartando.

3.5 Creación de nuevas generaciones a partir del cruce de cromosomas

El cruce de cromosomas de una población tiene como objetivo crear cromosomas hijos que “hereden” las características de los padres.

Como lo demuestran Alcaraz et al (2003) el cruce de cromosomas no implica solamente la combinación de información entre estos sino que al final esta sea beneficiosa logrando que a medida que transcurran las generaciones se obtengan mejores individuos. Un cruce de cromosomas mal diseñado se convierte en una mutación como lo expresan los autores.

El cruce de cromosomas padre y madre que utilizará el Algoritmo Genético propuesto será el mismo utilizado por Alcaraz et al (2003). El cruce de cromosomas a implementar es una modificación del cruce denominado “por dos puntos”.

Inicialmente se escogen de la población los cromosomas padres y madres a cruzar. La estrategia de definición de los cromosomas padre y madre es un híbrido entre la estrategia elitista y el método de la ruleta. La estrategia elitista tiene como principio que el 10% de los mejores padres forman el 50% de la población de hijos. El otro 50% de la población de hijos es generada por el 90% restante de la población de padres. Definidas los grupos de padres que se cruzarán para crear a los hijos, los individuos padre y madre se escogen utilizando el método de la ruleta en el cual se da una mayor probabilidad de escogencia a aquellos padres con mejor función objetivo.

Escogidos los cromosoma padre y madre que se cruzaran para formar los cromosomas hijo e hija, se definen aleatoriamente dos números enteros $K1$ y $K2$ donde $K1 < K2 < N-1$ siendo N el numero de actividades a programar incluyendo las actividades ficticias. A continuación se presenta un ejemplo del cruce de cromosomas tomando como base la representación de los cromosomas Padre y Madre que se puede observar en la figura 4.

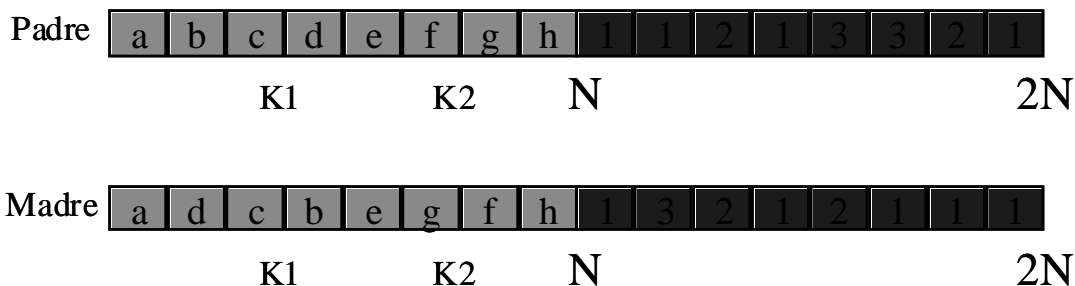


Figura 4. Representación de cromosomas Padre y Madre

En el cromosoma hijo la información almacenada en las $K1$ primeras posiciones será la misma que se encuentra en las $K1$ primeras posiciones del cromosoma madre. La información almacenada en las $K1$ primeras posiciones del cromosoma hija será la misma que se encuentra en las $K1$ primeras posiciones del cromosoma padre. En la figura 5 se puede observar como se almacenó la información en las primeras $K1$ posiciones de los cromosomas Hijo e Hija.

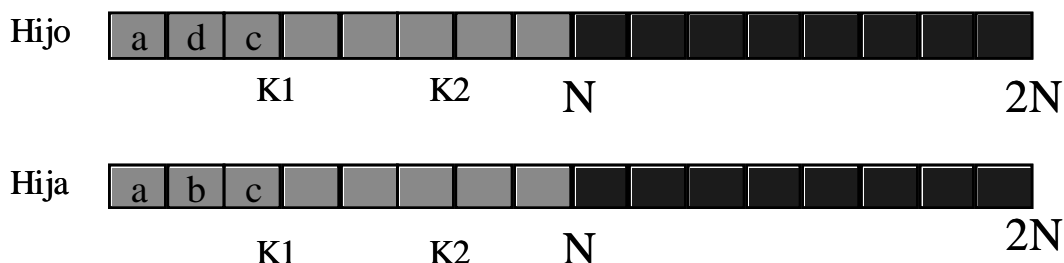


Figura 5. Información de las $K1$ posiciones de los cromosomas Hijo e Hija

La información almacenada entre las posiciones $K1 < j \leq K2$ del cromosoma hijo se tomará haciendo un barrido del cromosoma padre teniendo en cuenta la posición relativa de las actividades y verificando que la actividad que se encuentre en el cromosoma padre no se encuentre ya en el cromosoma hijo. En el caso del cromosoma hija el procedimiento es similar al realizado con el cromosoma padre. La diferencia radica que el barrido de las actividades se hará con el cromosoma madre. En la figura 6 se puede observar como se almacenó la información en las posiciones $K1 < j \leq K2$ de los cromosomas Hijo e Hija.

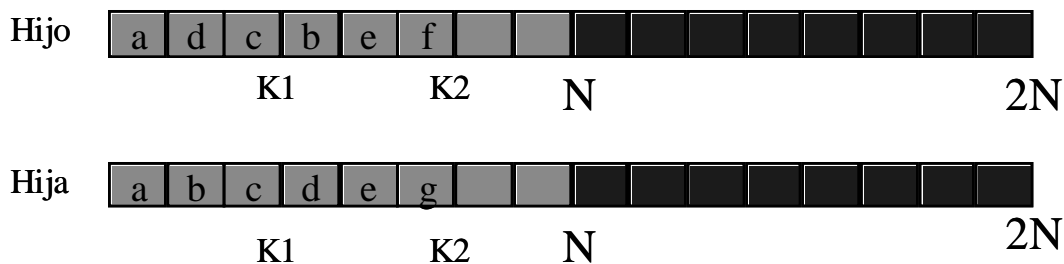


Figura 6. Información de las posiciones $K1 < j \leq K2$ de los cromosomas Hijo e Hija

La información almacenada en las posiciones $j > K2$ del cromosoma hijo se tomará del cromosoma madre siguiendo el mismo procedimiento que se realizó con anterioridad con el cromosoma padre para el caso de $K1 < j \leq K2$. La información almacenada en las posiciones $j > K2$ del cromosoma hija se tomará del cromosoma padre siguiendo el mismo procedimiento que se realizó con anterioridad con el cromosoma madre para el caso de $K1 < j \leq K2$. En la figura 7 se puede observar como se almacenó la información en las posiciones $j > K2$ de los cromosomas Hijo e Hija.

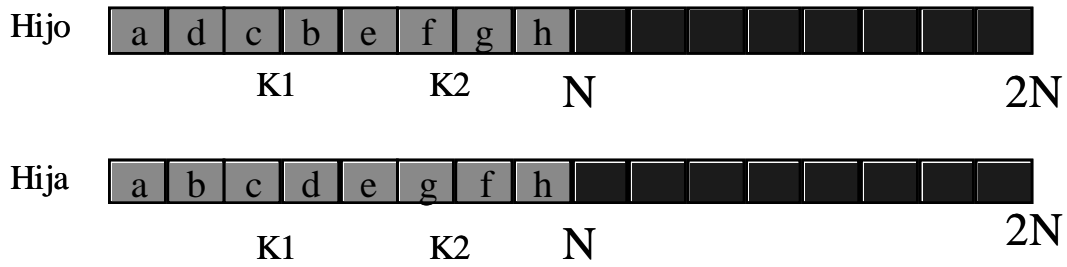


Figura 7. Información de las posiciones $j > K2$ de los cromosomas Hijo e Hija

Hay que tener en cuenta que en los cromosomas hijos e hijas las actividades provenientes de los cromosomas padre y madre conservaran los mismos modos como se puede ver en la grafica 8. Esto a diferencia del cruce de cromosomas que propone Hartmann (2001) en donde se presentan cambios en los modos asignados a las actividades al pasar del cromosoma padres a los hijos.

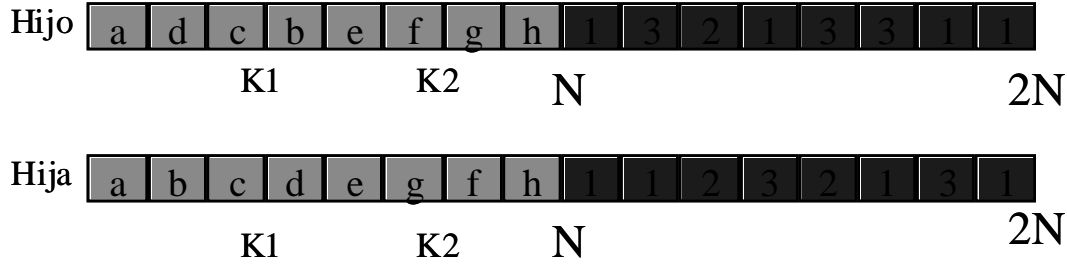


Figura 8. Información de las posiciones $j > N$ de los cromosomas Hijo e Hija

Finalmente la nueva población se obtendrá de los mejores I individuos entre la población de padres y la población de hijos en donde I es el tamaño predefinido que tendrá la población.

El criterio de parada del algoritmo genético será el número de generaciones obtenidas. La mejor solución será el primer individuo de la última generación.

3.6 Mutación de cromosomas

La mutación de los individuos de una población busca obtener nuevas soluciones que no sería posible generar a partir del cruce de cromosomas Alcaraz et al (2003). En el Algoritmo Genético propuesto se presentan dos tipos de mutación, las cuales afectarán a los individuos con una probabilidad definida previamente.

El primer tipo de mutación consiste en escoger aleatoriamente dos actividades del cromosoma con el fin de intercambiarlas de posiciones. Los modos en que son ejecutadas estas actividades se conservan por lo que también se intercambian posiciones entre ellos. Con el fin de garantizar que se cumplan las restricciones de secuencia se debe verificar que entre las actividades escogidas aleatoriamente y las actividades que se encuentre ubicadas en posiciones intermedias entre estas no haya relaciones de precedencia

El segundo tipo de mutación consiste en escoger aleatoriamente una actividad del cromosoma con el fin de cambiar el modo en que será ejecutada.

3.7 Procedimiento de reemplazo de modos

Con el fin de evitar que el Algoritmo Genético converja rápidamente a óptimos locales se ha introducido un procedimiento en el cual a la mitad de la población que represente a los peores individuos se le realizará el cambio de modo a todas las actividades. El Algoritmo Genético propuesto garantiza que ninguna de las actividades de los individuos que se vean afectados por este procedimiento permanecerán con el mismo modo en que se venían ejecutando.

4. REPROGRAMACIÓN DE PROYECTOS BAJO EL MODELO MMRC PSP

Como se ha expresado con anterioridad en este documento se genera la necesidad de reprogramar las actividades de un proyecto en el momento en que la ejecución de alguna o un grupo de ellas se ve afectada por una interrupción. Teniendo en cuenta las características del problema MMRC PSP se pueden presentar interrupciones como la llegada de una nueva actividad a ejecutar súbitamente, la salida de funcionamiento de uno de algún modo por un período de tiempo, la escasez o no disponibilidad de recursos renovables durante la ejecución de las actividades o la reducción en la disponibilidad de recursos no renovables.

En la investigación realizada se plantea como accionador del proceso de reprogramación la llegada de una nueva actividad la cual no estaba contemplada dentro del programa inicial de ejecución de actividades.

Esta interrupción al programa original puede generar que se detenga la ejecución de algunas actividades que desde este instante se denominarán actividades interrumpidas. Por otra parte la interrupción también podría generar impacto en aquellas actividades cuyo porcentaje de ejecución es 0%. Estas actividades se denominarán actividades no iniciadas. La actividad que arribe se denominará actividad entrante. El modo de ejecución de la actividad entrante será el modo 1, aunque este en realidad será un modo ficticio ya que la duración y el consumo de recursos renovables y no renovables de esta actividad estará predefinido.

La actividad entrante tendrá prioridad de ejecución sobre las actividades interrumpidas y las actividades no iniciadas. Esto quiere decir que se deberán disponer los recursos necesarios para ejecutar la actividad entrante así originalmente se haya planeado su utilización para ejecutar actividades interrumpidas y/o no iniciadas. De igual manera las actividades interrumpidas tendrán prioridad de ejecución con respecto a las actividades no iniciadas.

Si una actividad es interrumpida solo se retomará la reprogramación del porcentaje de ejecución que hacia falta antes que llegara la actividad entrante. Abumaizar et al (1997) indican que este tipo de interrupción de las actividades se denomina "interruption-resume". Los autores plantean que la duración de la actividad remanente será igual al tiempo total de procesamiento de la actividad menos el tiempo procesado antes de la interrupción. En el caso especial del problema MMRC PSP y como supuesto de esta investigación se define que toda actividad interrumpida deberá reprogramarse teniendo en cuenta el modo original que se definió al programarla.

Teniendo en cuenta las características de la reprogramación planteadas, en esta investigación se plantean dos técnicas que permitan acomodar la actividad entrante tratando de evitar cambios en el programa original.

La primera de ellas es una heurística basada en la propuesta realizada por Abumaizar et al (1997) denominada Heurística Affected Operation Rescheduling.

La heurística tiene como objetivo acomodar cualquier interrupción del programa original modificando los tiempos de inicio de algunas actividades hacia adelante la cantidad mínima requerida con el fin de que se preserve la secuencia del programa original

La otra técnica que se utilizará en esta investigación para reprogramar proyectos es un Algoritmo Genético basado en el planteado para la programación.

4.1 Heurística para la reprogramación de proyectos bajo el esquema MMRCPS

Basado en el trabajo de Abumaizar et al (1997) en el cual sólo se reprograman los trabajos afectados por la interrupción, se plantea una heurística que reprogramme proyectos bajo el modelo MMRCPS cuando se genera la llegada de una nueva actividad solamente atrasando aquellas actividades interrumpidas e iniciadas que se vean afectadas directamente o indirectamente por la interrupción.

Se sugieren dos tipo de causas por las que una actividad pueda verse afectada por la llegada de otra. La primera causa es que debido a la llegada de la actividad entrante se realice un consumo inesperado de recursos renovables que no permita la ejecución de las actividades programadas inicialmente.

Otra causa es que alguna actividad predecesora se retrase (por insuficiencia de recursos o por que la que la nueva actividad tiene prioridad en la ejecución) lo que impacta directamente en los tiempos de inicio de la actividad sucesora.

El objetivo de la heurística planteada es analizar la situación de cada una de las actividades interrumpidas e iniciadas teniendo en cuentas las posibles causas por las que podrían verse afectadas con el fin de definir si el tiempo de inicio de estas actividades debe ser atrasado.

A continuación se plantea un diagrama que representa el proceso de reprogramación realizado por la heurística.

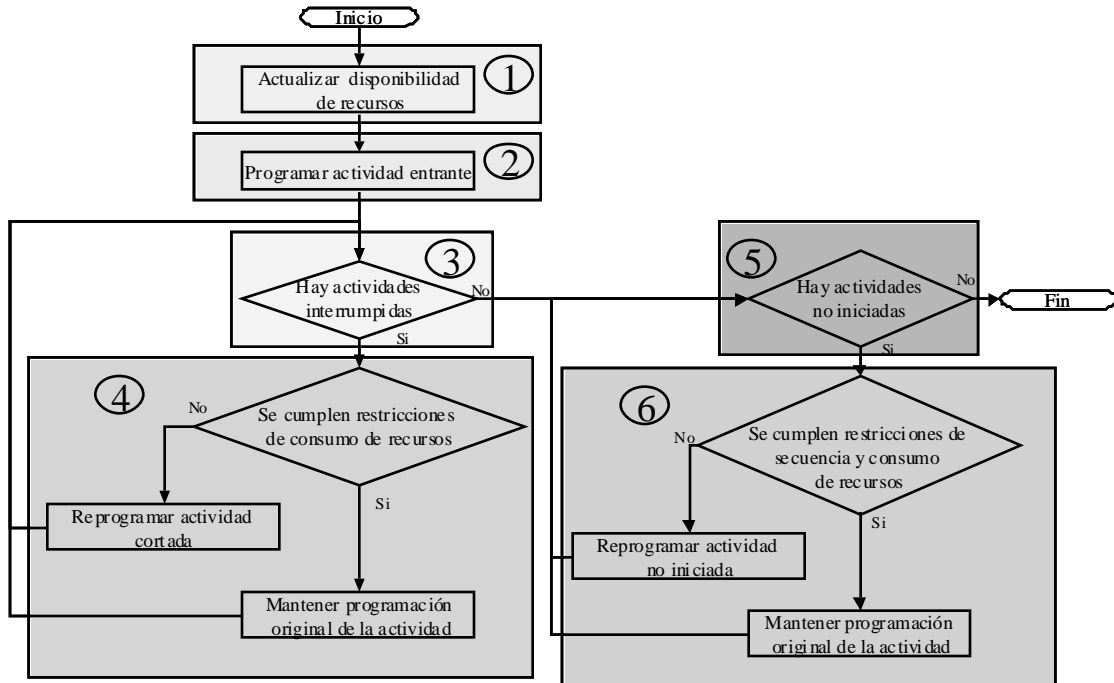


Figura 9. Diagrama proceso de reprogramación realizado por la heurística

Con el fin de detallar los pasos para implementar la heurística propuesta se plantea el siguiente ejemplo de un programa de ejecución de actividades de un proyecto. En la figura 10 se puede ver el programa original base del ejemplo. En esta gráfica el eje horizontal t corresponde a la duración de las actividades. El eje vertical r corresponde al consumo del recurso renovable por parte de cada una de las actividades.

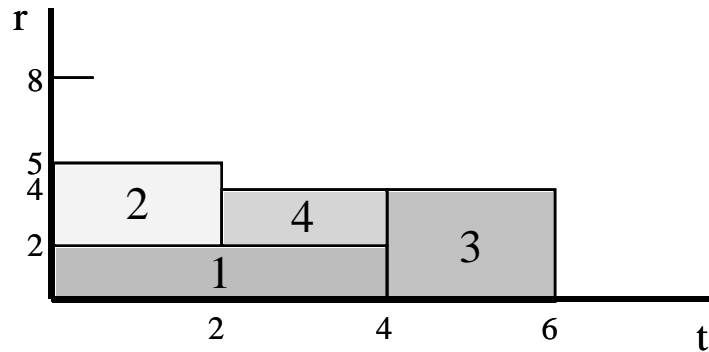


Figura 10. Programación original del proyecto

El programa planteado se ve interrumpido por la llegada de la actividad entrante “G” en el instante $t=1$. La duración y el consumo del recurso renovable por parte de cada una de las actividades del ejemplo, incluyendo la actividad entrante se puede observar en la siguiente tabla 2. Además se incluye las sucesoras por restricción de cada actividad.

Actividad	Duración	Consumo de recurso renovable r	Actividades sucesoras
1	4	2	3
2	2	3	4
3	2	4	---
4	2	2	---
G	2	4	---

Tabla 2. Información sobre las actividades interrumpidas, no iniciadas y generadas a reprogramar

Generada la interrupción se plantea la heurística de reprogramación

- (1) Desde el instante en que se genera la interrupción se permite la disponibilidad total de los recursos renovables. En el ejemplo, este paso equivale a eliminar todas las actividades del diagrama Gantt programadas desde $t \geq$ instante en que arriba la actividad entrante, como se puede observar en la figura 11.

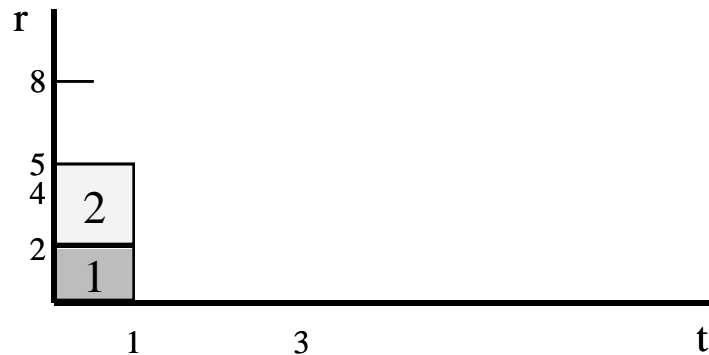


Figura 11. Eliminación actividades del diagrama Gantt

- (2) Teniendo disponibles los recursos renovables a su máxima capacidad se programa la actividad entrante. En el caso del ejemplo se programa la actividad G, como se puede ver en la figura 12.

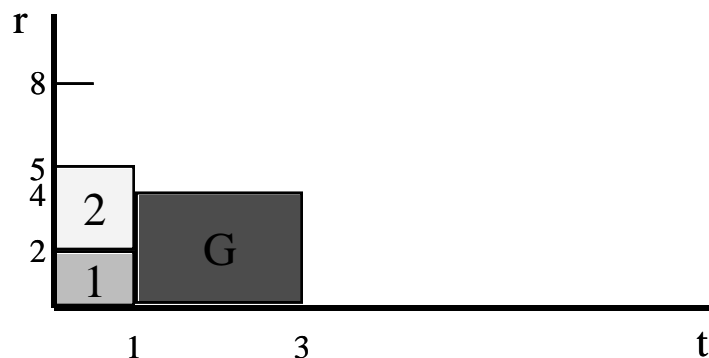


Figura 12. Programación actividad entrante G urgente

- (3) Se identifican si hay actividades interrumpidas. En el ejemplo las actividades 1 y 2 están en ejecución en el instante que arriba la actividad

entrante. Si no hay actividades interrumpidas se verifica si existen actividades sin iniciar (ver numeral 5 de esta sección)

- (4) Teniendo en cuenta el porcentaje de ejecución previo al arribo de la actividad entrante original se realiza el análisis de reprogramación de las actividades interrumpidas. En el caso del ejemplo, el porcentaje de ejecución de la actividad 2 es del 50% mientras que el porcentaje de ejecución de la actividad 1 es del 25%, por lo tanto la prioridad de entrada al programa la tiene la actividad 2.

Si después de programada la actividad entrante hay suficientes recursos renovables para ejecutar la actividad interrumpida no se modifica la programación original de esta. Por ejemplo, la programación de la ejecución de la actividad 2 no se modifica ya que se cuenta con la suficiente disponibilidad de recursos para programarla sin retrasar su tiempo de finalización.

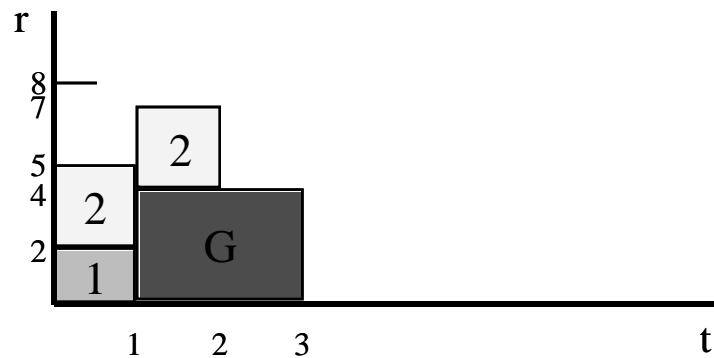


Figura 13. Programación actividad interrumpida 2

Si después de programada la actividad entrante no hay suficientes recursos renovables para ejecutar la actividad interrumpida se retrasa el tiempo de inicio de esta hasta el instante que se cuente con los recursos necesarios para su desarrollo. La heurística propuesta garantiza que los recursos sean suficientes desde el instante en que inicia la actividad hasta que esta finalice. Como se puede ver en el siguiente grafico el tiempo de inicio de la actividad 1 se retrasa hasta $t=2$ ya que en el intervalo $1 > t < 2$ no había la disponibilidad suficiente del recurso renovable R para su ejecución.

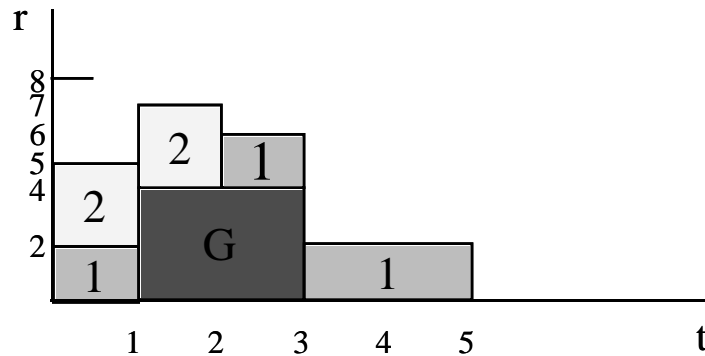


Figura 14. Programación actividad interrumpida 3.

- (5) Si no existen actividades interrumpidas o la totalidad de las actividades de estas fueron reprogramadas se identifican si hay actividades sin iniciar.
- (6) Con base en la prioridad de programación dada por el problema original (entiéndase solución del Algoritmo Genético propuesto) se realiza el análisis de reprogramación de las actividades sin iniciar. En el ejemplo primero se realizará la actividad 3 y luego la actividad 4.

Si no hay los suficientes recursos renovables disponibles y/o no se cumple la restricción de secuencia debido al atraso del tiempo de finalización de la actividad precedente por restricción, se modifica el tiempo de inicio original de la actividad no iniciada hasta que se cumplan las restricciones mencionadas. Por ejemplo, el tiempo de inicio de la actividad 3 es $t=4$. Dado que el tiempo de finalización de la actividad 1 luego de reprogramada es $t=5$ se atrasa el tiempo de inicio de la actividad 3 hasta este instante. Como se observa en el gráfico en los instantes $5 > t < 7$ hay los suficientes recursos para reprogramar la actividad 3.

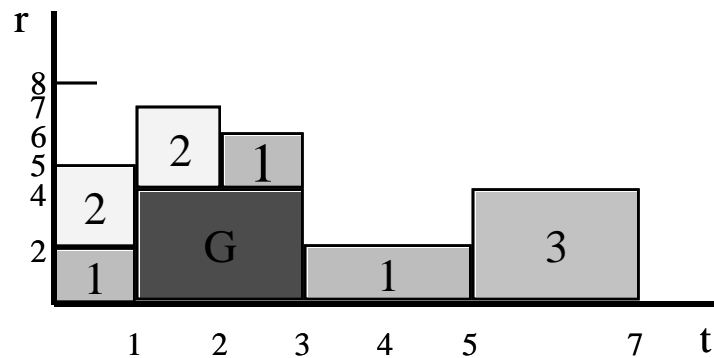


Figura 15. Programación actividad no iniciada 3.

Si la programación original de la actividad no iniciada cumple con las restricciones de secuencia y disponibilidad de recursos renovables esta no se modifica. En el caso del ejemplo la programación de la actividad 4 no sufre modificación ya que el máximo tiempo de finalización de sus actividades predecesoras no cambia y que además durante su ejecución ($2 > t < 4$) se cuenta con los suficientes recursos renovables disponibles.

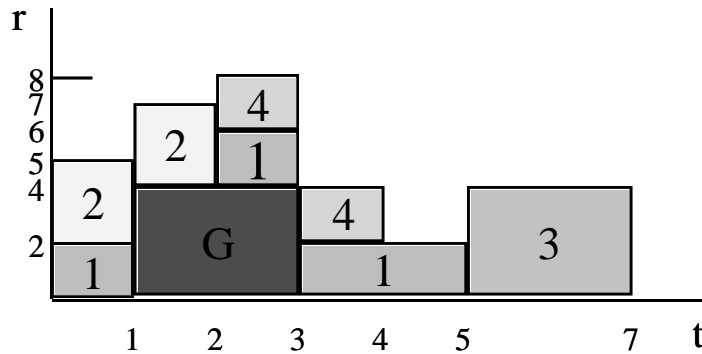


Figura 16. Programación actividad no iniciada 4.

Se puede observar en el gráfico anterior que el nuevo tiempo de finalización del proyecto es $t=7$ y que las actividades 1 y 3 sufrieron modificaciones en sus tiempos de inicio y finalización.

Además de lo expuesto sobre el procedimiento que sigue la heurística de reprogramación propuesta se debe aclarar que esta no realiza ningún tipo de validación sobre el cumplimiento de la restricción de disponibilidad de recursos no renovables. Se entiende que dado que la actividad entrante no estaba planificada pierde validez el restringir el proyecto por la capacidad de recursos no renovables. Los proyectos reales se detienen cuando se consumen la totalidad de recursos hasta que haya un reaprovisionamiento de estos. De igual manera uno de los objetivos principales de esta heurística es mantener estable el programa original evitando los cambios en los tiempos y modos de ejecución de las actividades.

4.2 Algoritmo Genético para la reprogramación de proyectos bajo el esquema MMRCPS

Con base en el Algoritmo Genético planteado para programar proyectos se plantea un nuevo algoritmo que permita reprogramar proyectos bajo las condiciones estipuladas al inicio del numeral 4. Se debe hacer énfasis en que la actividad entrante tiene prioridad de ejecución sobre las actividades interrumpidas y las no iniciadas. De igual manera las actividades interrumpidas tienen prioridad de ejecución sobre las actividades no iniciadas.

En comparación con el Algoritmo Genético para programación de proyectos, el Algoritmo Genético de reprogramación tiene ajustes en la representación de la solución, en la forma de cruce entre cromosomas, en la mutación de cromosomas y en el procedimiento de reemplazo de modos.

4.2.1 Ajuste en el cromosoma

El cromosoma que se utilizará para representar la solución del problema reprogramado consta de $2N$ posiciones en donde N corresponde a la suma de las actividades interrumpidas, las no iniciadas y la actividad entrante.

Como se puede ver en la siguiente gráfica, basada en el programa original del ejemplo de la heurística, la primera posición del cromosoma corresponderá a la actividad entrante. En las posiciones desde 2 hasta C se almacenarán las actividades interrumpidas, en donde C es el número de actividades interrumpidas. Las posiciones desde $C+1$ hasta N corresponderán a las actividades no iniciadas.



Figura 17. Representación de individuos para el Algoritmo Genético de reprogramación

De igual manera se define para cada una de las actividades el modo en que se ejecutará. La actividad entrante tendrá por defecto el modo 1 aunque este sea ficticio ya que la duración y el consumo de recursos renovables y no renovables de esta estarán predefinidos.

Los modos de las actividades interrumpidas no podrán ser modificados con el fin de mantener el curso normal de ejecución de la actividad con respecto al programa original. Por otra parte el Algoritmo Genético propuesto permite que los modos de las actividades no iniciadas puedan modificarse a diferencia de la heurística propuesta con anterioridad.

4.2.2 Cruce entre cromosomas

Teniendo en cuenta la representación del cromosoma planteado en el numeral 4.2.1 se plantean ajustes en la manera de cruzar los cromosomas padres para formar cromosomas hijos.

En el Algoritmo Genético propuesto para la reprogramación no se permite que se realice cruce entre las actividades interrumpidas y no iniciadas (hay que recordar que la actividad entrante siempre estará en la primera posición).

A diferencia de la heurística de reprogramación la prioridad de ejecución de las actividades interrumpidas la dará el Algoritmo Genético. Se permitirá cruce entre actividades interrumpidas siempre y cuando se cumplan las restricciones de secuencia.

Por otra parte podrá haber intercambio de posiciones entre actividades no iniciadas siempre y cuando se cumplan las restricciones de precedencia.

Finalmente se utilizará la técnica de cruce por un solo punto con el fin de intercambiar las actividades interrumpidas y no iniciadas entre los cromosomas padre y madre.

4.2.3 Ajuste en la mutación de cromosomas

Como se mencionó anteriormente no se podrá cambiar el modo de ejecución establecido en el programa original para las actividades interrumpidas. Por lo tanto en el Algoritmo Genético propuesto solo se realiza la mutación de las actividades no iniciadas.

4.2.4 Procedimiento de reemplazo de modos

Como en el caso anterior solamente se reemplazaran los modos de las actividades no iniciadas.

5. RESULTADOS COMPUTACIONALES

En esta sección se presentarán los resultados computacionales de la programación y reprogramación de proyectos enmarcados en el modelo MMRCPS.

Teniendo en cuenta los objetivos específicos de la investigación se ha planteado la necesidad de realizar un diseño de experimento con el fin de definir los datos de entrada paramétricos del Algoritmo Genético para la programación de proyectos y el desarrollo de un estudio para comparar los resultados del Algoritmo Genético para la reprogramación de proyectos versus la técnica heurística planteada.

Los Algoritmos Genéticos para la programación y reprogramación de proyectos fueron codificados y compilados en Visual Basic para Microsoft Excel. Los experimentos fueron realizados en un computador personal Pentium IV con procesador de 1600 Hz y 1 Gb de memoria RAM. Los problemas que se tomaron para realizar la investigación fueron obtenidos de la librería de problemas de programación de proyectos de la Universidad de Kiel. Estas instancias están disponibles en la dirección electrónica <http://www.bwl.uni-kiel.de/Prod/psplib>. Estas instancias han sido utilizadas por autores como Alcaraz et al (2003), Hartmann (2001), etc.

5.1 Configuración computacionales del Algoritmo Genético para la programación de proyectos

Para realizar la configuración del Algoritmo Genético para la programación de proyectos se ha definido un diseño de experimentos para lo cual se realizaron los siguientes pasos.

- **Comprensión y planteamiento del problema**

Como se ha definido con anterioridad el objetivo de este diseño de experimentos inicial es configurar el Algoritmo Genético que se utilizará para programar y reprogramar proyectos enmarcados en el modelo MMRCPS.

Específicamente se desea definir el tamaño de la población, la probabilidad de mutación de los cromosomas, el número de iteraciones que se ejecutarán y el porcentaje de ejecución del Algoritmo Genético desde donde se iniciará el proceso de reemplazo de los modos de los peores individuos de la población.

- **Elección de factores y niveles**

En la siguiente tabla se presentan los factores que variarán en el experimento y los niveles específicos en los cuales se hará el experimento, todo esto agrupado en tratamientos.

FACTOR	NIVEL
Tamaño de la población	25
	50
Probabilidad de mutación	0,2
	0,4
Cambios en los modos de la población	50%<
	75%<
Iteraciones	100
	200

Tabla 3. Factores y niveles del diseño de experimentos

- **Selección de la variable de respuesta**

Las variables de respuesta del diseño de experimentos a realizar serán los tiempos de terminación de los proyectos programados o C_{max} y el tiempo de ejecución del Algoritmo Genético propuesto medido en segundos.

- **Elección del diseño de experimentos**

El diseño de experimentos que se utilizará en esta etapa de la investigación es el Diseño Aleatorizado por Bloques Completos.

Este tipo de diseño de experimentos se caracteriza en que todos los tratamientos son probados en cada uno de los bloques. Si se usa este diseño, los bloques forman una unidad experimental más homogénea en la cual comparar los tratamientos. Esta estrategia de diseño mejora efectivamente la precisión en la comparación al eliminar la variabilidad entre los tratamientos. El orden en que los tratamientos son probados en cada bloque se determina aleatoriamente.

TRATAMIENTOS				
Número	Tamaño	Probabilidad de mutación	Cambio en modos	Iteraciones
1	26	0,2	0,5	100
2	26	0,2	0,5	200
3	26	0,2	0,75	100
4	26	0,2	0,75	200
5	26	0,4	0,5	100
6	26	0,4	0,5	200
7	26	0,4	0,75	100
8	26	0,4	0,75	200
9	50	0,2	0,5	100
10	50	0,2	0,5	200
11	50	0,2	0,75	100
12	50	0,2	0,75	200
13	50	0,4	0,5	100
14	50	0,4	0,5	200
15	50	0,4	0,75	100
16	50	0,4	0,75	200

Tabla 4. Tratamientos del diseño de experimentos

En el caso específico del diseño de experimentos propuestos se definen 16 tratamientos que corresponden a la combinación de factores en sus diferentes niveles. Los tratamientos definidos se pueden observar a continuación:

Se definieron 9 bloques correspondientes a 9 instancias o problemas seleccionados aleatoriamente de las disponibles en la dirección electrónica <http://www.bwl.unikel.de/Prod/psplib>.

- **Realización del experimento**

Se realizaron tres experimentos con el fin de definir la mejor configuración del Algoritmo Genético para programar proyectos de 16, 20 y 30 actividades.

Con el fin de obtener los resultados necesarios para determinar la mejor configuración del Algoritmo Genético de programación se realizaron tres corridas del algoritmo con los parámetros de cada uno de los tratamientos para los 9 bloques. En total para cada tamaño de proyecto se realizaron 432 corridas.

- **Análisis de datos y conclusiones**

Los datos obtenidos en el Diseño de experimentos son analizados utilizando el Test de Friedman. El Test de Friedman es el análogo no paramétrico del Análisis de Varianza planteándose en ambos casos la misma hipótesis nula (las medias de los tratamientos son iguales). La ventaja del Test de Friedman sobre el Análisis de Varianza es que en este no se tienen en cuenta los supuestos de normalidad de la población que implica la aleatoriedad y normalidad de los residuos y la varianza constante entre los tratamientos. Milton (2003)

A continuación se presenta el estadístico de prueba del Test de Friedman.

$$S = \left[\frac{12}{b \cdot k \cdot (k + 1)} \sum_{i=1}^k T_i^2 \right] - 3 \cdot b \cdot (k + 1)$$

donde

S = estadístico de prueba

b = número de bloques

k = número de tratamientos

T_i = totales por tratamiento

Como lo menciona Milton (2003) el estadístico de Friedman es una medida de las desviaciones de los totales de los rangos de los tratamientos observados respecto a su valor esperado. Calculado este estadístico se compara con el valor crítico de la distribución ji-cuadrado con k-1 grados de libertad con nivel de significancia del 0,05%. La hipótesis nula no se acepta con valores de *S* que excedan los valores obtenidos para la distribución ji-cuadrado.

En la siguiente tabla se presentan los valores obtenidos del estadístico de Friedman a partir de los datos del diseño de experimentos para proyectos de 16, 20 y 30 actividades.

Tamaño del proyecto	Estadístico de Friedman
16 actividades	2897,43
20 actividades	5066,36
30 actividades	8573,07

Tabla 5. Estadísticos de Friedman para los diferentes tamaños de proyectos

El valor crítico de la distribución ji-cuadrado con k-1 grados de libertad y un nivel de significancia del 0,05% es 24,99. Dado que los estadísticos de Friedman superan la valor crítico de la distribución ji-cuadrado puede concluirse no hay evidencia suficiente para aceptar que la media de los tratamientos son iguales.

Teniendo en cuenta esta conclusión se escogerán los mejores tratamientos para determinar la configuración del Algoritmo Genético de programación mediante los gráficos de los intervalos de las medias de los tratamientos a un nivel de confianza del 95%. En el anexo 1 se podrán encontrar los gráficos mencionados.

Teniendo en cuenta estos gráficos se seleccionan a criterio del investigador el mejor tratamiento para cada tamaño de proyecto del grupo que conforman aquellos con las medias más bajas y que a partir de la gráfica se puede inferir no tiene diferencia estadística. Entre los criterios que se utilizaron para escoger el mejor tratamiento se encuentra su media con respecto al Cmax y el tiempo promedio de ejecución. En la siguiente tabla se presenta este grupo de tratamientos por tamaño de proyecto y el tratamiento escogido para definir la configuración del Algoritmo Genético.

Tamaño del proyecto	Tratamientos sin diferencia estadística con respecto a la media	Tratamiento escogido
16 actividades	16,13,14,11,10,9,15,4,6,8,12	9
20 actividades	16,14,12,15,10,6,13,2,5,7,9	5
30 actividades	15,13,14,16,12,6,9,10,8	15

Tabla 6. Selección mejores tratamientos del diseño de experimentos

En las tablas 7, 8 y 9 se presenta la configuración del Algoritmo Genético para cada tamaño de proyecto a partir de los tratamientos escogidos.

Tamaño de la población	Probabilidad de mutación	Cambio en modos	Iteraciones
50	0,2	0,5	100

Tabla 7. Configuración Algoritmo Genético para proyectos de 16 actividades.

Tamaño de la población	Probabilidad de mutación	Cambio en modos	Iteraciones
26	0,4	0,5	100

Tabla 8. Configuración Algoritmo Genético para proyectos de 20 actividades.

Tamaño de la población	Probabilidad de mutación	Cambio en modos	Iteraciones
50	0,4	0,75	100

Tabla 9. Configuración Algoritmo Genético para proyectos de 30 actividades

5.2 Resultados computacionales del Algoritmo Genético para la programación de proyectos

Definidos los parámetros iniciales del Algoritmo Genético que programará proyectos de 16, 20 y 30 actividades se programan 52 instancias de las lecturas y se comparan los resultados obtenidos por investigaciones anteriores. En los anexos 5, 6 y 7 se pueden observar los resultados obtenidos para cada una de las 52 instancias programadas de los proyectos de 16, 20 y 30 actividades respectivamente. La siguiente tabla resume los resultados obtenidos por el Algoritmo Genético propuesto para cada una de las instancias.

Instancias	% Desviación del mejor resultado	Máxima desviación	% de factibilidad	% de optimalidad
J16	1.14	11.9%	100	75
J20	3.08	14.58%	100	59.61
J30	2.00	12.82%	100	--

Tabla 10. Resultados de la ejecución del Algoritmo Genético

En la tabla que se encuentra a continuación se comparan los resultados obtenidos por el Algoritmo Genético propuesto contra los óptimos de las instancias y los resultados obtenidos por el Algoritmo Genético de Alcaraz et al (2003) y el algoritmo de Enfriamiento Simulado de Jozefowska (2001).

Autor	INSTANCIAS		
	J16	J20	J30
Berrio (2006)	1.14	3.08	2.00
Alcaraz(2003)	1.12	1.91	--
Jozefowska (2001)	4.07	6.74	--

Tabla 11. Comparación resultados obtenidos con el Algoritmo Genético propuesto versus algoritmos de otros autores

La tabla anterior muestra que aunque el Algoritmo Genético propuesto no supera en ninguno de los casos los resultados obtenidos por Alcaraz et al (2003) este es competitivo ya que en promedio la desviación del óptimo de los resultados obtenidos para cada una de las instancias es menor al 5%.

Para el caso de los proyecto de 30 actividades (J30), Hartmann (2001) comenta que no se han encontrado soluciones óptimas. Para subsanar este inconveniente se han utilizado técnicas heurísticas las cuales han definido las mejores respuestas encontradas hasta el momento y que han servido como base para realizado esta comparación. A continuación se presenta una tabla donde se comparan los resultados obtenidos por el Algoritmo Genético propuesto contra el Algoritmo Genético de Hartmann para la programación de proyectos de 30 actividades.

Autor	% Desviación del mejor resultado	Maxima desviación	% de factibilidad
Berio (2006)	2	12.82%	100
Hartmann (2001)	16.93	151.9%	86.3

Tabla 12. Comparación resultados obtenidos con el Algoritmo Genético propuesto versus algoritmos de otros autores para problemas de 30 actividades.

Como se puede observar el Algoritmo Genético propuesto supera en todos los aspectos al Algoritmo Genético propuesto por Hartmann para programar proyectos de 30 actividades.

En cuanto a los tiempos de ejecución del Algoritmo Genético propuesto en la siguiente tabla se muestran los promedios para cada una de las instancias ejecutadas.

Instancias	Tiempo promedio de ejecución (seg)
J16	12.86
J20	8.97
J30	34.78

Tabla 13. Comparación tiempos de ejecución obtenidos con el Algoritmo Genético propuesto versus algoritmos de otros autores para problemas de 30 actividades.

5.3. Comparación resultados reprogramación de proyectos utilizando un Algoritmo Genético versus una técnica heurística

Teniendo claro que el Algoritmo Genético propuesto en esta investigación es competitivo con respecto a los Algoritmos propuestos por otros autores para programar proyectos enmarcados en el problema MMRCPS se procede en esta etapa de la investigación a comparar los resultados de reprogramar proyectos utilizando un Algoritmo Genético versus la técnica heurística que solo reprograma aquellas actividades que se ven afectadas por la interrupción del programa original. Como se mencionó con anterioridad la reprogramación de un proyecto se ejecutará cuando llegue a este una actividad la cual no estaba planeada y que tenga prioridad de ejecución sobre las demás actividades del proyecto.

Con el fin de realizar el diseño de la ejecución de los algoritmos propuestos y comparar los resultados obtenidos por cada uno de estos se han definido variables dependientes e independientes de la experimentación. Las variables independientes corresponden a aquellos factores o datos de entrada del problema de reprogramación que se modificarán para llevar a cabo el estudio. Las variables dependientes corresponden a los indicadores que permitirán medir la estabilidad del proyecto después que este haya sido reprogramado.

- Variables independientes

Con el fin de obtener la mayor cantidad posible de casos de reprogramación de proyectos se han identificado tres variables que se modificarán a diferentes niveles con el fin de realizar el estudio:

Tiempo de ocurrencia de la reprogramación: Se entiende como el instante en que la actividad entrante no planeada deba iniciar su ejecución. Para esta variable se diseñaron dos niveles: temprano y tarde. Estos niveles están definidos en intervalos de porcentajes del Cmax del proyecto original. En la tabla 14 se puede observar los intervalos planteados.

Duración: Corresponde al tiempo que consume la ejecución de la actividad entrante. Al igual que para la variable anterior, se han diseñado dos niveles de “Duración”: corta y larga. Estos niveles están definidos en intervalos de porcentajes del Cmax. En la tabla 14 se puede observar los intervalos planteados.

Tamaño de problema: Esta definida como la cantidad de actividad o tareas que hacen parte del proyecto. En este caso se manejarán los niveles de esta variable que se utilizaron durante estudio de la programación de proyectos. En la tabla 14 se puede observar los niveles planteados para esta variable.

VARIABLE INDEPENDIENTE	NIVEL
Tiempo de ocurrencia	Temprano [0,05Cmax;0,4Cmax]
	Tarde [0,6Cmax;0,9Cmax]
Duración	Corta [0,1Cmax;0,2Cmax]
	Larga [0,25Cmax;0,35Cmax]
Tamaño del problema	16 actividades
	20 actividades
	30 actividades

Tabla 14. Variables independientes y sus niveles

- Variables dependientes

Algunos autores han planteado la necesidad de medir la estabilidad del proyecto después que este ha sido reprogramado. Basado en el trabajo Abumaizar et al (1997) se han planteado los indicadores Porcentaje de cambio en el Cmax del proyecto y Desviación en tiempos. Igualmente en esta investigación se propone la utilización del indicador Desviación en modos de ejecución.

Porcentaje de cambio en el Cmax: Se define como el cambio porcentual entre el Cmax de proyecto original y el del proyecto reprogramado. Este indicador está clasificado dentro de los indicadores de eficiencia por Abumaizar et al (1997). A continuación se plantea la formulación matemática del indicador.

$$C \text{ max cambio} \% = \frac{C \text{ max reprogramado} - C \text{ max original}}{C \text{ max original}} * 100$$

donde

Cmax reprogramado: corresponde a la tardanza ponderada total de problema reprogramado.

Cmax original: corresponde a la tardanza ponderada total del problema original.

Cmax cambio%: porcentaje de cambio entre la solución del problema original y la solución del problema reprogramado.

Desviación en tiempos: En esta investigación el indicador de desviación en tiempos medirá los cambios entre las fechas de inicio y finalización de las actividades entre la solución original y la solución reprogramada.

$$DT = DTI + DTF$$

donde

DTI: corresponde a las sumatoria de las desviaciones en los tiempos de inicio de las actividades.

DTF: corresponde a la sumatoria de la desviación en los tiempos de finalización de las actividades.

Desviación en modos de ejecución: En esta investigación se propone este indicador de estabilidad como referente para medir el impacto que tiene la reprogramación de un proyecto en la planificación de los modos en que se ejecutan cada una de las actividades de este. En la vida real, por ejemplo, el cambio en los modos de un proyecto equivaldría a cambiar los funcionarios o empresas encargadas de ejecutar tareas en proyecto informáticos o de construcción respectivamente. Este indicador estará dado en cantidad de actividades del proyecto que fueron sujetas a cambio de modo de ejecución.

Tiempos computacionales: Este indicador medirá el tiempo en segundo que transcurre durante la ejecución del Algoritmo Genético y la heurística de reprogramación.

Definidas las variables independientes y dependientes que harán parte del estudio se procede a la ejecución de este. Para tal fin se escogen aleatoriamente 10 instancias de proyectos de 16, 20 y 30 actividades obtenidas de la dirección electrónica <http://www.bwl.unikiel.de/Prod/psplib>. En total son 30 instancias las cuales se programarán utilizando el Algoritmo Genético diseñado para la programar proyectos (Ver sección 3.) y luego se reprogramarán utilizando el Algoritmo Genético de reprogramación (Ver sección 4.2) y la heurística propuesta en esta investigación (Ver sección 4.1) teniendo en cuenta los diferentes niveles de las variables independientes que conforman los datos de duración y ocurrencia de la interrupción del programa original. Se debe tener claro que con el fin de realizar la comparación del desempeño de los algoritmos de reprogramación propuestos la ejecución de estos se realizará para resolver el mismo problema de reprogramación (el programa original y la actividad entrante tendrán las mismas características).

En los anexos 5, 6 y 7 se pueden observar los resultados de la reprogramación de cada uno de los problemas de 16, 20 y 30 actividades los cuales fueron interrumpidos por una actividad entrante no planeada cuyos datos de duración y tiempos de ocurrencia están dados por la combinación de los diferentes niveles de las variables independientes.

A continuación se presenta el análisis de la reprogramación tomando como base las variables dependientes planteadas con anterioridad.

Porcentaje de cambio en el C_{max}

Con el fin de iniciar el análisis de los resultados de la reprogramación de proyectos tomando como base el indicador de Porcentaje de cambio en el C_{max} se presenta la siguiente tabla resumen de resultados.

Tamaño del proyecto	% Cambio en el C _{max}	
	Heurística	Algoritmo Genético
16 actividades	3,16	11,40
20 actividades	4,30	9,29
30 actividades	2,23	7,27
Promedio	3,23	9,32

Tabla 15. Resultados indicador Porcentaje de cambio en el C_{max}

Es claro que la heurística planteada en esta investigación es más eficiente que el Algoritmo Genético cuando se trata de reprogramar proyectos enmarcados en el modelo MMRCPS. Los resultados de la heurística son en promedio tres veces mejores que los arrojados por el Algoritmo Genético. De igual manera hay que resaltar que para ninguno de los diferentes tamaños de proyectos el porcentaje de cambio del C_{max} después de reprogramar el proyecto con la heurística superó el 5%.

Aunque hay evidencia de la superioridad del desempeño de la técnica heurística versus el Algoritmo Genético en la reprogramación de proyectos teniendo como base el indicador de Porcentaje de cambio en el C_{max} del proyecto es necesario tener en cuenta que en la ejecución de algunas instancias, el desempeño del Algoritmo Genético fue superior o igual al desempeño de la técnica heurística. A continuación se presenta una comparación detallada del desempeño del Algoritmo Genético versus la técnica heurística.

Tamaño del proyecto	Frecuencia (instancias)	
	C _{max} (AG) < C _{max} (Heu)	C _{max} (AG) = C _{max} (Heu)
16 actividades	5	8
20 actividades	11	10
30 actividades	6	9
Total	22	27

Tabla 16. Instancias en las que el desempeño del Algoritmo Genético fue superior o igual a la técnica heurística

En la tabla 16 se puede observar que en 22 y 27 instancias (en total se ejecutaron 120 instancias sumando los tres tamaños de problemas) el Algoritmo Genético obtiene resultados superiores o iguales a los obtenidos por la técnica heurística respectivamente.

En la siguiente tabla se presenta información sobre el desempeño de cada uno de los algoritmos de reprogramación (AG y Heurística) empleados con respecto al C_{max} del programa original.

Tamaño del proyecto	Frecuencia (instancias)		
	$C_{max}(AG) < C_{max} \text{ Original}$	$C_{max}(AG) = C_{max} \text{ Original}$	$C_{max}(\text{Heu}) = C_{max} \text{ Original}$
16 actividades	0	10	33
20 actividades	1	6	22
30 actividades	1	8	30

Tabla 17. Desempeño algoritmos de reprogramación versus C_{max} del programa original

En la segunda columna de la tabla se puede observar como en dos instancias ejecutadas el C_{max} del proyecto reprogramado con el Algoritmo Genético es inferior al C_{max} del proyecto original. Esto se presenta debido a que el Algoritmo Genético planteado en esta investigación tiene la posibilidad de realizar cambio en los modos de las actividades no iniciadas al momento de generarse la interrupción del programa original. Esto conlleva a que si la programación original no es óptima es posible que al reprogramar el proyecto se obtenga un tiempo de finalización menor al del proyecto original. El hecho de que este caso solo se haya presentado en el 1,6% de las instancias se convierte en una prueba del buen desempeño del Algoritmo Genético que se planteó para programar proyectos.

En las columnas 3 y 4 de la tabla se muestra como en el 20% y 71% de las instancias no se presentaron cambios en el C_{max} del proyecto después de haber sido reprogramado con el Algoritmo Genético y la técnica heurística respectivamente. Esto a las claras es una muestra más del desempeño superior de la técnica heurística sobre el Algoritmo Genético en cuanto al indicador en estudio.

Finalmente en las siguientes tablas se presentan los resultados obtenidos por cada una los algoritmos de reprogramación planteados teniendo en cuenta los niveles de variables independientes.

Niveles variables independientes		% Cambio en el C_{max} Proyectos 16 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	1,61	7,88
Temprano	Larga	5,88	9,80
Tarde	Corta	1,56	12,65
Tarde	Larga	3,60	15,27

Tabla 18. Resultados de la reprogramación de proyectos de 16 actividades teniendo en cuenta los diferentes niveles de las variables independientes.

Niveles variables independientes		% Cambio en el C_{max} Proyectos 20 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	5,34	9,98
Temprano	Larga	4,15	9,90
Tarde	Corta	2,12	7,51
Tarde	Larga	5,58	9,76

Tabla 19. Resultados de la reprogramación de proyectos de 20 actividades teniendo en cuenta los diferentes niveles de las variables independientes.

Niveles variables independientes		% Cambio en el C _{max} Proyectos 30 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	2,40	4,88
Temprano	Larga	5,33	7,96
Tarde	Corta	0,79	6,89
Tarde	Larga	0,40	9,34

Tabla 20. Resultados de la reprogramación de proyectos de 30 actividades teniendo en cuenta los diferentes niveles de las variables independientes.

Del análisis de las tablas anteriores se puede llegar a las siguientes conclusiones:

- En la medida que haya menos posibilidades de cambio en el programa original, esto es que las interrupciones se generen lo más tarde posible y estas sean de más duraderas, el desempeño de la técnica heurística mejora.
- Se obtiene mejores desempeños reprogramando proyectos con Algoritmos Genéticos entre más temprana sea la ocurrencia y corta la duración de la interrupción. Esto se debe a que hay mayores posibilidades de cambios en el programa original.

Desviación en tiempos

Con el fin de iniciar el análisis de los resultados de la reprogramación de proyectos tomando como base el indicador de desviación de tiempos se presenta la siguiente tabla resumen de resultados.

Tamaño del proyecto	Desviación de tiempos (días)	
	Heurística	Algoritmo Genético
16 actividades	12,45	42,70
20 actividades	21,85	69,45
30 actividades	24,45	91,35
Promedio	19,58	67,83

Tabla 21. Resultados indicador Desviación de tiempos

La información registrada en la tabla anterior indica que en promedio los proyectos reprogramados utilizando la heurística que solo reprograma aquellas actividades que se ven afectadas por la interrupción son más estables que aquellos que son reprogramados utilizando el Algoritmo Genético propuesto en esta investigación. Para todos los tamaños de proyectos la desviación en tiempo de las tareas es tres veces superior en el Algoritmo Genético versus la heurística.

En las siguientes tablas se presentan los resultados obtenidos por cada uno de los algoritmos de reprogramación planteados teniendo en cuenta los niveles de variables independientes para el indicador de Desviación de tiempos.

Niveles variables independientes		Desviación de tiempos (días) Proyectos de 16 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	9,3	50,4
Temprano	Larga	30,8	63,8
Tarde	Corta	30,8	63,8
Tarde	Larga	6,7	30,1

Tabla 22. Resultados de la reprogramación de proyectos de 16 actividades para el indicador Desviación de tiempos

Niveles variables independientes		Desviación de tiempos (días) Proyectos 20 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	35,1	100,8
Temprano	Larga	33,3	100,8
Tarde	Corta	6,7	29,4
Tarde	Larga	12,3	46,8

Tabla 23. Resultados de la reprogramación de proyectos de 20 actividades para el indicador Desviación de tiempos

Niveles variables independientes		Desviación de tiempos (días) Proyectos 30 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	42,5	136,5
Temprano	Larga	49,4	141,7
Tarde	Corta	4,6	45,8
Tarde	Larga	1,3	41,4

Tabla 24. Resultados de la reprogramación de proyectos de 30 actividades para el indicador Desviación de tiempos

Se puede observar en las tablas anteriores que para ambos algoritmos de reprogramación en la medida que la ocurrencia de la interrupción sea más tarde y su duración sea mas larga la desviación en los tiempos disminuye a excepción de casos puntuales. Esto es debido a que en la medida que la interrupción cumpla las condiciones mencionadas hay una menor probabilidad de que la mayoría de las actividades del proyecto se ven afectadas por esta.

Desviación en modos de ejecución

Con el fin de iniciar el análisis de los resultados de la reprogramación de proyectos tomando como base el indicador de Desviación en modos de ejecución se presenta la siguiente tabla resumen de resultados.

Tamaño del proyecto	Desviación de modos (número de modos)	
	Heurística	Algoritmo Genético
16 actividades	0	1,98
20 actividades	0	2,48
30 actividades	0	4,13
Promedio	0	2,86

Tabla 25. Resultados indicador Desviación de modos de ejecución

Es claro que este indicador solo puede ser medido a partir del desempeño del Algoritmo Genético ya que la técnica heurística planteada en esta investigación no permite que se realicen cambios de modos durante la reprogramación de proyectos. Como se podía esperar a medida que aumenta el número de actividades del proyecto a reprogramar también aumenta el número de actividades que cambian de modo de ejecución.

En las siguientes tablas se presentan los resultados obtenidos por el Algoritmo Genético de reprogramación teniendo en cuenta los niveles de variables independientes para el indicador de Desviación de modos de ejecución.

Niveles variables independientes		Desviación de modos (# de modos) Proyectos 16 actividades
Ocurrencia interrupción	Duración interrupción	Algoritmo Genético
Temprano	Corta	3
Temprano	Larga	3,6
Tarde	Corta	0,6
Tarde	Larga	0,7

Tabla 26. Resultados de la reprogramación de proyectos de 16 actividades para el indicador Desviación de modos de ejecución

Niveles variables independientes		Desviación de modos (# de modos) Proyectos 20 actividades
Ocurrencia interrupción	Duración interrupción	Algoritmo Genético
Temprano	Corta	3,4
Temprano	Larga	3,8
Tarde	Corta	1,2
Tarde	Larga	1,5

Tabla 27. Resultados de la reprogramación de proyectos de 20 actividades para el indicador Desviación de modos de ejecución

Niveles variables independientes		Desviación de modos (# de modos) Proyectos 30 actividades
Ocurrencia interrupción	Duración interrupción	Algoritmo Genético
Temprano	Corta	6,7
Temprano	Larga	7,1
Tarde	Corta	1,3
Tarde	Larga	1,4

Tabla 28. Resultados de la reprogramación de proyectos de 30 actividades para el indicador Desviación de modos de ejecución

Analizando los datos registrados en las tablas anteriores se puede observar que sin importar la ocurrencia de la interrupción o en el caso de esta investigación la llegada de una nueva actividad al proyecto con prioridad de ejecución el Algoritmo Genético entrega mejores resultados en la reprogramación teniendo en cuenta el indicador de Desviación de modos si la duración de la interrupción es corta.

Tiempos computacionales

En la siguiente tabla se presentan los tiempos de ejecución promedio de los algoritmos propuestos de reprogramación para cada uno de los tamaños de proyectos.

Tamaño del proyecto	Tiempos de ejecución (seg)	
	Heurística	Algoritmo Genético
16 actividades	0,07	9,89
20 actividades	0,05	6,89
30 actividades	0,13	21,41
Promedio	0,08	12,73

Tabla 29. Resultados indicador Tiempos de ejecución de algoritmos

Es claro el superior desempeño de la heurística de reprogramación sobre el Algoritmo Genético lográndose que en todas las instancias tiempos menores a 1 seg. Los tiempos del Algoritmo Genético de reprogramación siguen el mismo comportamiento que los obtenidos para el Algoritmo Genético para la programación de proyectos.

En las siguientes tablas se presentan los resultados obtenidos por cada uno de los algoritmos de reprogramación planteados teniendo en cuenta los niveles de variables independientes para el indicador de Tiempos de ejecución.

Niveles variables independientes		Tiempos de ejecución (seg) Proyectos de 16 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	0,08	13,72
Temprano	Larga	0,10	14,56
Tarde	Corta	0,06	5,68
Tarde	Larga	0,04	5,62

Tabla 30. Resultados de la reprogramación de proyectos de 16 actividades para el indicador Tiempos de ejecución.

Niveles variables independientes		Tiempos de ejecución (seg) Proyectos 20 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	0,07	10,15
Temprano	Larga	0,07	10,56
Tarde	Corta	0,03	3,42
Tarde	Larga	0,03	3,42

Tabla 31. Resultados de la reprogramación de proyectos de 20 actividades para el indicador Tiempos de ejecución.

Niveles variables independientes		Tiempos de ejecución (seg) Proyectos 30 actividades	
Ocurrencia interrupción	Duración interrupción	Heurística	Algoritmo Genético
Temprano	Corta	0,19	31,71
Temprano	Larga	0,19	36,39
Tarde	Corta	0,08	9,50
Tarde	Larga	0,06	8,03

Tabla 32. Resultados de la reprogramación de proyectos de 30 actividades para el indicador Tiempos de ejecución.

En el caso del Algoritmo Genético se puede observar que en la medida en que la interrupción se genere tarde los tiempos computacionales disminuyen ostensiblemente. Esto se debe a que menos actividades se ven impactadas por la reprogramación por lo que el tamaño del cromosoma de representación de individuos disminuye.

6. CONCLUSIONES

Dada la importancia que ha logrado la gestión de las actividades de cualquier organización a través de proyectos se ha hecho indispensable la generación de métodos computacionales que permitan al gerente de proyectos realizar la programación y reprogramación de las actividades de este.

Diversos autores han estudiado el tema de la reprogramación de tareas en áreas como la manufactura flexible, programación de la producción en una sola máquina, programación de la producción en maquina paralelas, etc. En cuanto a la reprogramación de proyectos pocas investigaciones se han realizado, en especial de proyectos enmarcados en el problema MMRCPS el cual es objeto de esta investigación.

Teniendo en cuenta los algoritmos computacionales diseñados por otros autores resolver problemas de reprogramación de tareas se ha diseñado en esta investigación un Algoritmo Genético y una técnica heurística que permitan reprogramar proyectos enmarcados en el problema MMRCPS cuando una nueva actividad con prioridad de ejecución interrumpe la programación del proyecto.

Inicialmente se ha diseñado un Algoritmo Genético que permita realizar la programación inicial del proyecto. Este algoritmo es la base del Algoritmo Genético que reprogramará el proyecto. Comparando los resultados obtenidos de la ejecución del Algoritmo Genético de programación con los encontrados en la literatura se puede concluir que éste es competitivo ya que aunque no superó para los proyectos de 16 y 20 actividades los resultados de la investigación de Alcaraz et al (2003) en promedio la desviación del óptimo para el total de instancias es menor del 5%. En el caso de los proyectos de 30 actividades el Algoritmo Genético propuesto obtiene resultados que se desvían un 2% del mejor resultado encontrado hasta el momento superando ampliamente los obtenidos por el algoritmo de Hartmann (2001) que obtiene una desviación del 16.93%.

Teniendo clara la competitividad del Algoritmo Genético planteado para la programación del proyecto se realizan ajustes y se desarrolla un diseño de experimentos que permita configurar el algoritmo Genético que reprogramara los proyectos. Por otra parte se diseña una técnica heurística que solo reprograma aquellas actividades que hayan sido afectadas directa o indirectamente por la interrupción generada por la llegada de una nueva actividad. Ambos métodos computacionales se comparan en ambientes de reprogramación en donde se modifican los tiempos de ocurrencia de la interrupción y la duración de esta. Igualmente se utilizan indicadores de eficiencia y estabilidad de la reprogramación como son el Porcentaje de cambio en el C_{max} y la Desviación en tiempos planteados por diferentes autores y se diseña en esta investigación un nuevo indicador denominado Desviación en modos de ejecución.

Revisando los resultados computacionales para el indicador de Porcentaje de cambio en el C_{max} se concluye que la técnica heurística de reprogramación

supera ampliamente en eficiencia al Algoritmo Genético de reprogramación con un porcentaje de cambio en C_{max} del programa original de 3,23% contra el 9,32 % obtenido por el Algoritmo Genético. De igual manera se debe resaltar el hecho de que el 1,6% de las instancias ejecutadas el Algoritmo Genético de reprogramación pudo disminuir el tiempo de finalización del proyecto original aunque se haya generado una interrupción. Esto se convierte en una oportunidad para los gerentes de proyectos reales ya que aunque la probabilidad es baja es posible que el Algoritmo Genético de reprogramación pueda mejorar las imperfecciones de la programación original. De igual manera en el 20% de las instancias el Algoritmo Genético de reprogramación incluye la nueva actividad sin afectar el tiempo de finalización del proyecto aunque este porcentaje es bajo compara con el 71% obtenido por la heurística lo que demuestra claramente el desempeño superior de esta.

En cuanto a la estabilidad del proyecto reprogramado se identifica que la heurística también supera al Algoritmo Genético de reprogramación dado que se obtiene en promedio una desviación de tiempos de 19,58 días en comparación con el 67,83 del Algoritmo Genético de reprogramación. También se concluye que a medida que el tamaño del proyecto aumenta, en ambos algoritmos la desviación en los tiempos después de reprogramando tiene una tendencia al crecimiento.

Los resultados obtenidos para el nuevo indicador planteado en esta investigación que mide la desviación en cuanto a la planificación original de los modos en que se ejecutarían las actividades indican que el Algoritmo Genético entrega mejores resultados en la medida que la duración de la interrupción sea corta. En el caso de este indicador no se obtienen resultados para la heurística ya que el diseño de esta no permite que se modifiquen los modos de ejecución de las actividades definidos en el programa original.

Finalmente y para corroborar el desempeño superior en cuanto a la reprogramación de proyectos enmarcados en el modelo MMRCPS de la heurística que solo reprograma la actividades afectadas directa o indirectamente por la interrupción se observa que los tiempos computaciones de ejecución de ésta son menores al segundo contra un promedio de 12,72 seg del Algoritmo Genético de reprogramación.

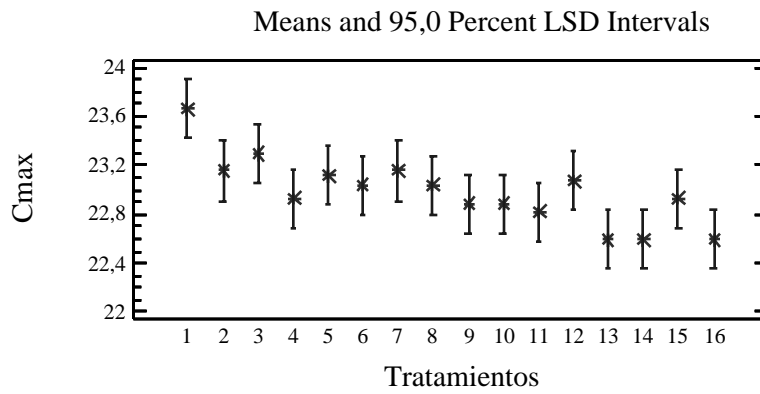
7. BIBLIOGRAFÍA

1. Abumazar, R., Svestka, J. Rescheduling job shops under random disruptions. *International Journal of Production*. Vol 37, No 7, pag 2065-2082. 1997.
2. Alcaraz, J., Maroto, C. & Ruiz, R. Solving the multi-mode resource-constrained Project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*. Vol 54. pag 614-626. 2003.
3. Bierwirth, C. Mattfeld, C. Production Scheduling and Rescheduling with Genetic Algorithms. *Evolutionary Computation* 7(1): pag 1-17. 1999.
4. Calhoun, K., Deckro, R. & Moore, J. Planning and re-planning in project and production scheduling. *The International Journal of Management Science*. pag 155-170. 2002.
5. Chase, R., Aquilano, N. & Jacobs, F. *Administración de Producción y Operaciones, Manufactura y Servicios*. Octava edición, Ed. McGraw Hill. Santa Fe de Bogotá, Colombia. 2001.
6. Demeulemeester, E. Herroelen, W. *Project scheduling with time windows – from the theory to applications*. 2002.
7. Hartmann, S. Project Scheduling with multiple modes: A Genetic Algorithm. *Annals of Operations Research*. Vol 102. pag 11-135. 2001.
8. Hartmann, S. *Project scheduling under limited resources: models, methods and applications*. Ed Springer. Berlín, Alemania. 1999.
9. Kwei, R., Shyu, Y. & Adiga, S. A heuristic rescheduling algorithm for computer-based production scheduling systems. *International Journal of Production*. Vol 31, No 8, pag 1815-1826. 1993.
10. Mason, C. Jin, S. & Wessels, C. Rescheduling strategies for minimizing total weighted tardiness in complex job shops. *International Journal of Production*. Vol 42, No 3, pag 613-628. 2004.
11. Milton, S. *Probabilidad y estadística: con aplicaciones para ingeniería y ciencias computacionales*. Ed Mc GrawHill. Ciudad de México, México. 2003.
12. Tavaréz, L. A review of the contribution of operational research to Project Management. *European Journal of Operational Research*. Vol 136. pag 1-18. 2002.
13. Vieira, G., Herrmann, J. & Lin, E. Rescheduling manufacturing systems: A framework of strategies, policies and methods. *Journal of Scheduling*. Vol 6. pag 39-62. 2003.

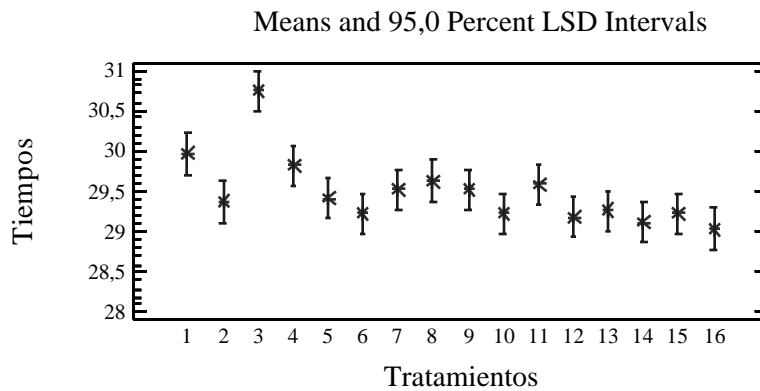
14. Wong, W., Leung, S. & Au, K. Real-time GA based rescheduling approach for the pre-sewing stage of an apparel manufacturing process. International Journal of Advanced Manufacturing Technologies. Vol 25. pag 180-188.2005.

ANEXO 1. GRÁFICOS INTERVALOS DE MEDIAS DE LOS TRATAMIENTOS PARA LOS DIFERENTES TAMAÑOS DE PROYECTOS

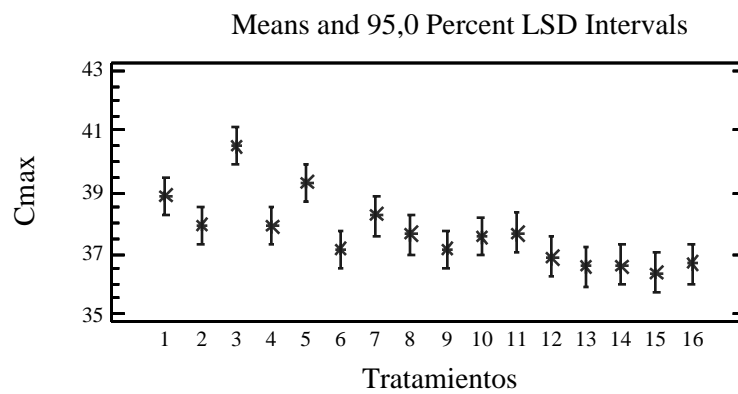
Proyectos de 16 actividades



Proyectos de 20 actividades



Proyectos de 30 actividades



ANEXO 2. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 16 ACTIVIDADES CON ALGORITMO GENÉTICO

Problema	Óptimo	Resultado Cmax				% Desviación promedio vs óptimo	Tiempos de ejecución (seg)			
		Iteración 1	Iteración 2	Iteración 3	Promedio		Iteración 1	Iteración 2	Iteración 3	Promedio
c1511_8	20	23	21	21	21,67	8,33	13,67	13,50	13,66	13,61
c1511_9	27	27	27	27	27	0	13,33	13,25	13,52	13,36
c1520_7	21	21	21	21	21	0	12,78	12,61	12,91	12,77
c1531_4	19	19	19	19	19	0	11,77	11,72	11,86	11,78
c1542_10	25	28	25	25	26	4	13,63	13,67	13,77	13,69
c1523_5	24	24	25	24	24,33	1,39	12,00	12,22	12,05	12,09
c1513_10	28	30	31	33	31,33	11,90	13,58	13,67	13,58	13,61
c1526_10	20	20	20	20	20	0	12,42	12,34	12,25	12,34
c1522_3	20	20	20	20	20	0	11,25	11,44	11,48	11,39
c1514_1	16	16	16	16	16	0	12,58	12,59	12,53	12,57
c1515_3	17	17	17	17	17	0	12,86	12,81	12,83	12,83
c1515_4	16	16	16	16	16	0	12,70	12,69	12,86	12,75
c1516_7	23	23	23	23	23	0	12,84	13,02	13,11	12,99
c1516_8	19	19	19	19	19	0	13,55	13,64	13,55	13,58
c1519_8	23	23	23	23	23	0	12,50	12,41	12,31	12,41
c1519_9	21	21	21	21	21	0	13,22	13,22	13,05	13,16
c1521_5	31	31	34	31	32	3,23	13,31	13,03	13,19	13,18
c1521_6	23	25	25	25	25	8,70	11,84	11,91	11,98	11,91
c1522_2	16	16	16	16	16	0	11,91	11,89	11,88	11,89
c1522_3	20	20	20	20	20	0	11,48	11,33	11,30	11,37
c1527_8	15	15	15	15	15	0	12,11	11,94	11,81	11,95
c1527_9	22	22	22	22	22	0	12,06	12,09	11,96	12,05
c1530_3	23	23	23	23	23	0	11,89	11,94	12,09	11,97
c1530_4	20	20	20	20	20	0	12,25	12,28	12,06	12,20
c1532_6	21	21	21	21	21	0	12,08	11,98	12,19	12,08
c1532_7	21	21	21	21	21	0	12,13	12,06	12,05	12,08
c1543_6	20	20	21	20	20,33	1,67	12,69	12,61	12,77	12,69
c1546_5	23	23	23	23	23	0	13,17	13,02	13,13	13,10
c1546_6	16	16	18	18	17,33	8,33	12,14	12,20	12,27	12,20
c1548_4	19	19	19	19	19	0	13,13	13,13	13,20	13,15
c1548_5	28	28	28	28	28	0,00	13,05	13,16	13,30	13,17
c1551_8	21	21	22	21	21,33	1,59	12,19	11,88	12,03	12,03
c1556_6	17	17	17	17	17	0	12,22	12,06	12,03	12,10
c1556_7	18	18	18	18	18	0	11,73	11,80	11,97	11,83
c1558_5	22	22	22	22	22	0	11,84	12,11	12,00	11,98
c1558_6	24	24	24	24	24	0	11,95	11,97	11,97	11,96
c1561_4	30	32	30	32	31,33	4,44	12,48	12,66	12,27	12,47
c1561_5	26	26	26	26	26	0	11,61	11,66	11,50	11,59
c1562_6	20	20	20	20	20	0	10,94	11,02	11,11	11,02
c1563_6	24	24	24	24	24	0	11,08	11,16	11,05	11,09
c1563_7	16	16	16	16	16	0	11,09	11,09	11,13	11,10
c1564_9	19	19	19	19	19	0	11,48	11,47	11,63	11,53
c1563_10	24	24	24	24	24	0	11,61	11,75	11,72	11,69
c1564_1	18	18	18	18	18	0	11,31	11,27	11,23	11,27
c1564_2	19	19	19	19	19	0	11,39	11,39	11,38	11,39
c1564_3	15	15	15	15	15	0	11,02	10,97	11,02	11,00
c1564_4	14	14	14	14	14	0	10,81	10,89	10,95	10,89
c1564_5	22	22	22	22	22	0	11,36	11,44	11,38	11,39
c1546_10	28	28	29	28	28,33	1,19	13,39	13,38	13,17	13,31
c1547_1	24	24	24	24	24	0	12,91	12,92	13,14	12,99
c1547_2	22	23	23	24	23,33	6,06	13,02	12,94	13,00	12,98
c1547_4	25	27	25	27	26,33	5,33	13,23	13,48	13,31	13,34

ANEXO 3. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 20 ACTIVIDADES CON ALGO RITMO GENÉTICO

Problema	Óptimo	Resultado Cmax				% Desviación promedio vs óptimo	Tiempos de ejecución (seg)			
		Iteración 1	Iteración 2	Iteración 3	Promedio		Iteración 1	Iteración 2	Iteración 3	Promedio
c2010_8	17	18	17	18	17,67	3,92	9,20	9,44	9,34	9,33
c2013_9	36	36	40	37	37,67	4,63	9,58	9,64	9,63	9,61
c2015_4	27	27	27	27	27	0	9,55	9,45	9,80	9,60
c2018_5	23	23	23	23	23	0	8,81	8,98	8,77	8,85
c2025_6	26	26	26	26	26	0	8,48	8,36	8,55	8,46
c2029_9	39	40	39	40	39,67	1,71	9,06	8,88	9,09	9,01
j2063_3	20	30	30	30	30	50	8,67	8,72	8,70	8,70
c2043_2	29	30	30	29	29,67	2,30	9,77	9,66	9,89	9,77
c2056_9	23	23	23	23	23	0	9,02	9,23	9,02	9,09
j2063_4	27	27	27	27	27	0	8,41	8,34	8,47	8,41
j2063_5	35	35	35	35	35	0	8,69	8,75	8,72	8,72
j209_1	28	30	28	32	30	7,14	9,47	9,55	9,33	9,45
j2031_1	25	25	25	25	25	0	8,56	8,59	8,66	8,60
c2018_6	21	21	21	21	21	0	8,66	8,73	8,58	8,66
j2012_9	32	32	32	32	32	0	10,14	9,92	9,97	10,01
j2012_10	20	20	20	20	20	0	9,03	8,83	8,86	8,91
j2013_1	27	28	29	30	29	7,41	9,39	9,55	9,61	9,52
j2031_3	23	25	25	25	25	8,70	8,52	8,55	8,64	8,57
j2013_3	45	45	45	48	46	2,22	10,30	10,64	10,08	10,34
j2031_4	26	29	28	27	28	7,69	8,67	8,78	8,70	8,72
j2018_2	27	27	27	27	27	0	8,91	8,97	8,83	8,90
j2018_3	20	22	22	20	21,33	6,67	8,94	8,98	8,89	8,94
j2018_4	21	23	23	22	22,67	7,94	8,56	8,63	8,80	8,66
j2018_5	23	23	23	23	23	0	8,83	8,78	9,05	8,89
j2018_6	21	21	21	21	21	0	8,41	8,67	8,63	8,57
j2020_8	17	18	17	17	17,33	1,96	8,97	8,70	8,64	8,77
j2020_9	34	34	34	34	34	0	9,42	9,27	9,45	9,38
j2020_10	21	21	21	21	21	0	8,59	8,63	8,69	8,64
j2060_4	24	24	24	24	24	0	8,48	8,42	8,42	8,44
j2060_5	21	21	21	21	21	0	8,72	8,53	8,67	8,64
j2063_8	21	21	21	21	21	0	8,84	8,73	8,73	8,77
j2063_6	29	29	29	29	29	0	8,59	8,66	8,55	8,60
j2063_7	36	36	36	36	36	0	8,53	8,56	8,59	8,56
j2021_10	28	28	28	28	28	0	8,61	8,64	8,67	8,64
j2022_1	27	27	28	28	27,67	2,47	8,44	8,44	8,41	8,43
j2022_2	30	33	35	34	34	13,33	8,86	8,67	8,81	8,78
j2060_3	23	23	23	23	23	0	9,22	9,14	9,27	9,21
j2022_4	26	28	29	29	28,67	10,26	9,00	8,89	8,89	8,93
j2060_2	22	22	22	22	22	0	8,67	8,72	8,69	8,69
j2044_10	26	26	26	26	26	0	9,45	9,23	9,61	9,43
j2018_7	25	25	25	25	25	0	8,83	9,05	8,91	8,93
j2018_8	29	29	29	29	29	0	8,80	8,94	9,13	8,95
j2057_7	26	26	27	27	26,67	2,56	8,97	8,86	8,94	8,92
j2045_4	32	36	38	36	36,67	14,58	9,30	9,75	9,80	9,61
j2045_5	38	39	39	39	39	2,63	9,91	9,91	9,98	9,93
j2045_6	36	36	36	37	36,33	0,93	9,22	9,59	9,77	9,53
j2059_8	22	23	22	22	22,33	1,52	8,77	8,75	8,67	8,73
j2059_9	27	27	27	27	27	0	8,92	9,16	8,91	8,99
j2059_10	26	26	26	26	26	0	8,81	9,00	8,88	8,90
j2060_1	18	18	18	18	18	0	8,69	8,58	8,72	8,66
j2060_2	22	22	22	22	22	0	8,75	8,86	8,78	8,80
j2063_1	27	27	27	27	27	0	8,59	8,59	8,61	8,60

ANEXO 4. RESULTADOS PROGRAMACIÓN DE PROYECTOS DE 30 ACTIVIDADES CON ALGORITMO GENÉTICO

Problema	Óptimo	Resultado Cmax				% Desviación promedio vs óptimo	Tiempos de ejecución (seg)			
		Iteración 1	Iteración 2	Iteración 3	Promedio		Iteración 1	Iteración 2	Iteración 3	Promedio
C3011_8	30	30	30	30	30	0	35,63	35,33	35,50	35,48
C3021_4	37	38	37	38	37,67	1,80	34,08	34,19	34,00	34,09
c3031_6	32	32	32	32	32	0	31,69	31,86	32,05	31,86
C3048_7	32	34	34	34	34	6,25	44,11	44,03	44,11	44,08
j3051_8	33	33	33	33	33	0	36,33	36,27	36,20	36,27
c3056_4	32	32	32	32	32	0	35,55	35,31	35,73	35,53
j3044_5	29	30	30	29	29,67	2,30	36,88	37,09	37,48	37,15
c3018_9	25	25	25	25	25	0	32,70	32,95	33,52	33,06
j3044_6	25	29	25	28	27,33	9,33	38,41	37,45	37,59	37,82
j3044_7	35	36	38	38	37,33	6,67	37,58	37,80	37,92	37,77
j3044_8	36	36	36	36	36	0	37,97	37,44	37,34	37,58
j3012_7	27	27	28	27	27,33	1,23	36,61	36,34	37,03	36,66
j3012_8	28	28	28	28	28	0	39,27	38,88	37,95	38,70
j3012_9	22	22	23	24	23	4,55	36,55	35,98	36,38	36,30
j3012_10	28	28	28	28	28	0	37,52	36,81	36,78	37,04
j3014_3	30	33	33	32	32,67	8,89	38,09	37,89	37,14	37,71
j3014_4	34	37	35	37	36,33	6,86	38,42	38,09	39,02	38,51
j3014_5	26	29	29	30	29,33	12,82	38,34	37,89	37,84	38,03
j3014_6	29	30	29	29	29,33	1,15	34,59	34,66	34,67	34,64
j3017_1	34	34	34	34	34	0	34,89	34,14	34,20	34,41
j3017_2	26	26	26	26	26	0	33,47	33,66	33,14	33,42
j3017_3	33	33	33	33	33	0	35,95	35,88	35,09	35,64
j3017_4	39	39	39	39	39	0	34,72	34,27	34,47	34,48
j3017_9	34	34	34	34	34	0	34,16	33,95	34,61	34,24
j3017_10	26	28	28	28	28	7,69	33,53	33,80	33,77	33,70
j3018_1	28	28	28	28	28	0	32,75	32,97	33,55	33,09
j3018_2	33	33	33	33	33	0	33,55	33,36	33,73	33,55
j3028_7	37	38	37	37	37,33	0,90	36,42	35,41	35,53	35,79
j3028_8	36	36	36	36	36	0	34,59	35,13	35,03	34,92
j3028_9	28	28	28	28	28	0	33,14	33,41	33,52	33,35
j3028_10	26	26	26	27	26,33	1,28	32,91	32,81	33,23	32,98
j3045_9	42	45	44	46	45	7,14	39,78	41,61	39,47	40,29
j3063_9	25	25	25	25	25	0	32,75	32,91	32,77	32,81
j3064_1	25	25	25	25	25	0	31,61	31,70	31,66	31,66
j3064_2	23	23	23	23	23	0	31,63	31,50	31,47	31,53
j3064_7	34	34	34	34	34	0	31,72	31,73	31,91	31,79
j3064_9	31	31	31	31	31	0	32,30	32,27	32,28	32,28
j3064_10	36	36	36	36	36	0	32,47	32,41	32,61	32,49
j3050_6	37	37	37	37	37	0	35,83	34,89	35,89	35,47
j3050_7	22	23	23	23	23	4,55	33,48	33,23	32,95	33,22
j3050_8	27	27	27	27	27	0	35,64	35,81	35,70	35,72
j3050_9	25	25	25	25	25	0	34,30	34,17	34,28	34,25
j3030_5	30	32	31	32	31,67	5,56	32,97	32,91	32,88	32,92
j3030_6	41	44	45	44	44,33	8,13	33,77	33,78	33,59	33,71
j3030_7	27	27	27	27	27	0	30,92	31,14	31,13	31,06
j3030_8	28	28	28	28	28	0	31,41	31,44	31,33	31,39
j3030_9	36	36	36	36	36	0	31,52	31,67	31,78	31,66
j3052_1	32	32	32	32	32	0	34,45	34,22	35,13	34,60
j3052_2	23	25	24	23	24	4,35	32,94	32,64	33,61	33,06
j3052_3	22	22	22	23	22,33	1,52	33,83	33,59	33,66	33,69
j3052_4	29	30	29	29	29,33	1,15	33,02	33,23	32,94	33,06
j3052_5	30	30	30	30	30	0	33,69	34,48	34,00	34,06

ANEXO 5. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 16 ACTIVIDADES

Características actividad entrante		Ocurrencia temprana y duración corta		Ocurrencia temprana y duración larga		Ocurrencia tardía y duración corta		Ocurrencia tardía y duración larga	
Problema	Indicadores	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético
1	% Cambio Cmax	0	0	0	7,41	0	0	0	3,70
	Desviación tiempos	14	27	0	71	0	7	0	7
	Cambio en modos	0	4	0	3	0	0	0	0
	Duración ejecución	0,09	15,92	0,09	16,06	0,14	4,63	0,03	4,78
2	% Cambio Cmax	11,11	0	5,56	16,67	5,56	11,11	16,67	16,67
	Desviación tiempos	40	26	36	75	11	32	15	45
	Cambio en modos	0	3	0	4	0	1	0	0
	Duración ejecución	0,11	16,33	0,08	15,55	0,13	7,47	0,05	7,59
3	% Cambio Cmax	0	23,53	0	5,88	0	23,53	0	23,53
	Desviación tiempos	0	74	32	33	0	16	0	22
	Cambio en modos	0	0	0	3	0	0	0	1
	Duración ejecución	0,06	10,81	0,08	13,19	0,03	4,08	0,03	4,19
4	% Cambio Cmax	0	23,53	0	17,65	0	11,76	0	5,88
	Desviación tiempos	9	107	0	87	0	22	0	13
	Cambio en modos	0	4	0	4	0	0	0	0
	Duración ejecución	0,08	13,11	0,08	13,25	0,05	5,03	0,03	5,16
5	% Cambio Cmax	0	0	0	0	0	0	0	0
	Desviación tiempos	0	9	0	4	2	7	0	4
	Cambio en modos	0	1	0	0	0	1	0	1
	Duración ejecución	0,06	8,55	0,06	8,83	0,03	4,81	0,05	4,89
6	% Cambio Cmax	0	10,53	0	15,79	0	5,26	0	15,79
	Desviación tiempos	0	72	0	52	0	12	0	15
	Cambio en modos	0	2	0	2	0	0	0	1
	Duración ejecución	0,08	12,13	0,08	15,33	0,06	5,75	0,02	4,17
7	% Cambio Cmax	0	8,70	0	8,70	0	17,39	4,35	26,09
	Desviación tiempos	0	63	0	74	0	35	6	46
	Cambio en modos	0	3	0	5	0	2	0	2
	Duración ejecución	0,11	16,52	0,09	17,11	0,05	6,77	0,05	6,02
8	% Cambio Cmax	0	12,5	20,83	4,17	0	25	0	25
	Desviación tiempos	0	101	86	104	0	40	0	40
	Cambio en modos	0	7	0	9	0	2	0	0
	Duración ejecución	0,08	15,02	0,20	16,11	0,03	4,91	0,05	5
9	% Cambio Cmax	5	0	15	0	10	15	15	10
	Desviación tiempos	30	20	72	44	17	43	46	31
	Cambio en modos	0	4	0	3	0	0	0	1
	Duración ejecución	0,08	12,83	0,09	13,86	0,05	6,56	0,05	7,58
10	% Cambio Cmax	0	0	17,39	21,74	0	17,39	0	26,09
	Desviación tiempos	0	5	82	94	0	51	0	78
	Cambio en modos	0	2	0	3	0	0	0	1
	Duración ejecución	0,09	15,97	0,09	16,30	0,03	6,75	0,05	6,83

ANEXO 6. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 20 ACTIVIDADES

Características actividad entrante		Ocurrencia temprana y duración corta		Ocurrencia temprana y duración larga		Ocurrencia tardía y duración corta		Ocurrencia tardía y duración larga	
Problema	Indicadores	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético
1	% Cambio Cmax	2,70	0,05	0	0	0	5,41	0	10,81
	Desviación tiempos	20	101	0	143	0	23	0	29
	Cambio en modos	0	8	0	5	0	2	0	2
	Duración ejecución	0,02	11,38	0,02	12,19	0,02	3,52	0,02	2,92
2	% Cambio Cmax	0	17,39	0	30,43	0	16,23	0	8,70
	Desviación tiempos	0	105	0	167	0	15	0	41
	Cambio en modos	0	1	0	3	0	4	0	3
	Duración ejecución	0,02	10,92	0,02	9,53	0,02	3,52	0,02	3,73
3	% Cambio Cmax	7,41	18,52	3,70	3,70	0	0	0	14,81
	Desviación tiempos	50	111	29	84	4	10	0	55
	Cambio en modos	0	2	0	5	0	1	0	1
	Duración ejecución	0,02	7,63	0,02	12,05	0,02	2,77	0,02	5,02
4	% Cambio Cmax	13,04	8,70	4,35	0	0	8,70	17,39	4,35
	Desviación tiempos	42	47	10	25	0	12	22	36
	Cambio en modos	0	3	0	5	0	1	0	1
	Duración ejecución	0,02	9,58	0,02	12,36	0,02	2,75	0,02	3,31
5	% Cambio Cmax	0	9,52	9,52	23,81	0	4,76	4,76	4,76
	Desviación tiempos	0	62	18	95	6	28	32	47
	Cambio en modos	0	3	0	2	0	0	0	1
	Duración ejecución	0,02	8,25	0,08	6,02	0,03	3,73	0,03	2,89
6	% Cambio Cmax	7,41	7,41	0	7,41	0	18,52	0	14,81
	Desviación tiempos	33	95	0	36	0	65	0	68
	Cambio en modos	0	6	0	4	0	2	0	2
	Duración ejecución	0,08	8,39	0,08	7,33	0,05	3,25	0,03	3,38
7	% Cambio Cmax	0	8,82	0	2,94	0	0	0	14,71
	Desviación tiempos	0	132	0	121	0	52	4	93
	Cambio en modos	0	4	0	6	0	0	0	3
	Duración ejecución	0,125	12,86	0,13	11,52	0,06	4,19	0,05	3,98
8	% Cambio Cmax	5,56	2,78	8,33	5,56	5,56	2,78	5,56	2,78
	Desviación tiempos	82	73	108	115	4	13	4	37
	Cambio en modos	0	2	0	3	0	0	0	0
	Duración ejecución	0,125	11,1875	0,14	13,03	0,03	2,81	0,03	2,94
9	% Cambio Cmax	12,5	21,88	15,63	15,63	15,63	18,75	28,13	21,88
	Desviación tiempos	104	225	168	159	53	73	61	59
	Cambio en modos	0	3	0	4	0	2	0	1
	Duración ejecución	0,14	11,19	0,14	12,09	0,06	4,42	0,03	3,17
10	% Cambio Cmax	4,76	4,76	0	9,52	0	0	0	0
	Desviación tiempos	20	57	0	63	0	3	0	3
	Cambio en modos	0	2	0	1	0	0	0	1
	Duración ejecución	0,11	10,16	0,11	9,53	0,05	3,20	0,05	2,81

ANEXO 7. RESULTADOS REPROGRAMACIÓN PARA PROYECTOS DE 30 ACTIVIDADES

Características actividad entrante		Ocurrencia temprana y duración corta		Ocurrencia temprana y duración larga		Ocurrencia tardía y duración corta		Ocurrencia tardía y duración larga	
Problema	Indicadores	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético	Resultado Heurística	Resultado Algoritmo Genético
1	% Cambio Cmax	0	15,63	0	9,38	0	0	0	6,25
	Desviación tiempos	0	195	0	189	0	7	0	90
	Cambio en modos	0	5	0	6	0	2	0	2
	Duración ejecución	0,20	37,91	0,20	38,16	0,08	9,61	0,06	9,88
2	% Cambio Cmax	3,70	11,11	3,70	22,22	0	7,41	0	11,11
	Desviación tiempos	58	186	10	259	0	27	0	36
	Cambio en modos	0	6	0	7	0	0	0	0
	Duración ejecución	0,17	31,28	0,20	40,91	0,06	7,36	0,06	6,58
3	% Cambio Cmax	0	0	8	4	0	16	4	20
	Desviación tiempos	14	52	48	124	8	50	13	98
	Cambio en modos	0	1	0	2	0	0	0	2
	Duración ejecución	0,17	25,72	0,17	29,31	0,09	12,72	0,09	12,92
4	% Cambio Cmax	0	2,44	21,95	4,88	4,88	2,44	0	7,32
	Desviación tiempos	6	233	163	158	12	79,00	0	45
	Cambio en modos	0	13	0	10	0	1	0	2
	Duración ejecución	0,20	35	0,17	28,72	0,06	8,78	0,05	5,56
5	% Cambio Cmax	7,14	0	0	7,14	0	17,86	0	0
	Desviación tiempos	64	83	2	75	0	65	0	19
	Cambio en modos	0	6	0	6	0	2	0	4
	Duración ejecución	0,17	26,27	0,19	35,84	0,08	9,81	0,06	8,91
6	% Cambio Cmax	0	9,09	9,09	9,09	3,03	3,03	0	12,12
	Desviación tiempos	0	176	175	151	10	22	0	26
	Cambio en modos	0	5	0	11	0	0	0	1
	Duración ejecución	0,14	20,45	0,20	41,19	0,05	4,58	0,05	7,75
7	% Cambio Cmax	0	4	0	4	0	0	0	0
	Desviación tiempos	8	69	0	126	0	8	0	8
	Cambio en modos	0	1	0	2	0	0	0	0
	Duración ejecución	0,17	31,55	0,19	34,06	0,05	7,20	0,06	7,31
8	% Cambio Cmax	0	3,57	0	7,14	0	14,29	0	14,29
	Desviación tiempos	0	87	0	129	0	37	0	32
	Cambio en modos	0	8	0	9	0	1	0	1
	Duración ejecución	0,22	38,39	0,19	36,63	0,08	7,16	0,06	7,33
9	% Cambio Cmax	13,16	0	10,53	0	0	7,89	0	10,53
	Desviación tiempos	275	178	96	102	16	143	0	11
	Cambio en modos	0	16	0	8	0	5	0	1
	Duración ejecución	0,25	41,41	0,19	37,52	0,16	18,66	0,05	5,42
10	% Cambio Cmax	0	2,94	0	11,76	2,94	0,03	0	11,76
	Desviación tiempos	0	106	0	104	0	20	0	49
	Cambio en modos	0	6	0	10	0	2	0	1
	Duración ejecución	0,19	29,11	0,20	41,56	0,08	9,09	0,06	8,63