

**PROGRAMACIÓN DE PROYECTOS DE CONSTRUCCIÓN DE EDIFICACIONES
MEDIANTE UN ALGORITMO GENÉTICO**

ANDRÉS CALDERÓN SAFFON

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE CIVIL
BOGOTÁ D.C.
2005**

**PROGRAMACIÓN DE PROYECTOS DE CONSTRUCCIÓN DE EDIFICACIONES
MEDIANTE UN ALGORITMO GENÉTICO**

ANDRÉS CALDERÓN SAFFON

**Trabajo de grado presentado como requisito
para optar al título de Ingeniero Civil**

**Directores:
SANDRA PATRICIA JARRO
JUAN PABLO CABALLERO**

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE
BOGOTÁ D.C.
2005**

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Bogotá, D.C. Junio de 2005

RESUMEN

La inversión en proyectos inmobiliarios implica un alto riesgo debido a las grandes sumas de dinero involucradas en las obras civiles. Luego de la crisis económica de los años noventa se han generado nuevas formas de negociación de los terrenos, lo cual tiene un impacto directo sobre la planeación de los proyectos. Este trabajo contempla dos escenarios enmarcados dentro de esta nueva realidad: uno condicionado por la capacidad de inversión o endeudamiento del constructor y otro en el cual se busca obtener la mayor rentabilidad sobre la inversión. Se desarrolla una herramienta computacional basada en un algoritmo genético que sirve de soporte para la toma de decisiones en la etapa de planeación de proyectos. Mediante el uso de la herramienta computacional se busca (1) ofrecer una programación de obra ajustada a los requerimientos del proyectista y (2) comprobar la aplicabilidad de los algoritmos genéticos en el caso colombiano.

Palabras clave: algoritmos genéticos, operadores genéticos, programación de obra, construcción, ingeniería civil, proyectos inmobiliarios, visual Basic, Colombia.

ABSTRACT

There is a high risk associated to the investment in the real state business due to the great amount of money involved in the construction activity. After the economic crisis occurred in the 90's some new ways of terrain trade have showed up, these have a direct impact over the project planning. This paper analyzes two scenarios in which this new reality is taken into account. The first one depends on the credit or investor's capacity. The second one seeks for obtaining the best reward upon the investment. The basis of this paper is the development of an application based on a genetic algorithm to support the decision making process in the project planning. The goals are (1) to offer a project planning adjusted to the project manager's requirements, and (2) to show and prove the application of a genetic algorithm in the solution of a typical construction case in Colombia.

Key words: genetic algorithms, genetic operators, project planning, construction, civil engineering, real state, visual basic, Colombia.

CONTENIDO

	pág.
INTRODUCCIÓN	14
OBJETIVOS	15
1. REVISIÓN BIBLIOGRÁFICA	17
1.1 ¿QUÉ SON LOS ALGORITMOS GENÉTICOS?	17
1.2 INTERCAMBIO TIEMPO-COSTO	21
1.3 LA CODIFICACIÓN DEL CROMOSOMA	22
1.4 EVALUACIÓN Y APTITUD	25
1.5 TÉCNICAS DE SELECCIÓN	26
1.5.1 Selección por torneo	29
1.5.2 Selección proporcional	30
1.5.3 Selección por ranking	34
1.5.4 Elitismo.	34
1.6 LA RECOMBINACIÓN	35
1.7 LA MUTACIÓN	37
1.8 LOS ESQUEMAS, ¿POR QUÉ FUNCIONAN LOS AGS?	37
2. DESCRIPCIÓN DEL PROBLEMA	41
2.1 IDENTIFICACIÓN DEL PROBLEMA	41
2.2 PLANTEAMIENTO DEL MODELO MATEMÁTICO	43
2.3 CONSIDERACIONES PREVIAS DEL MODELO MATEMÁTICO	48
2.4 PARTICULARIDADES DEL PROBLEMA PARA EL CASO COLOMBIANO	49
3. DESARROLLO DEL ALGORITMO GENÉTICO	51
3.1 CODIFICACIÓN DEL CROMOSOMA	51
3.1.1 Codificación plana con alelos consecutivos	51
3.1.2 Codificación plana con alelos no consecutivos	52
3.1.3 Codificación circular	52
3.1.4 Impacto de la codificación en el operador de recombinación	53
3.2 TÉCNICA DE SELECCIÓN	57

3.3 RECOMBINACIÓN	58
3.3.1 Recombinación con un solo punto de cruce y dos padres	59
3.3.2 Recombinación con dos puntos de cruce y dos padres	59
3.3.3 Recombinación con dos puntos de cruce y tres padres	60
3.4 MUTACIÓN	60
3.4.1 Reemplazo aleatorio	61
3.4.2 Incremento – decremento	61
3.4.3 Mutación Variable	61
3.5 SELECCIÓN DE LOS INDIVIDUOS QUE COMPONEN UNA NUEVA GENERACIÓN	61
3.5.1 Mejores padres mejores hijos	61
3.5.2 Elitismo	61
3.6 FUNCIÓN DE APTITUD	62
3.7 ¿POR QUÉ UN ALGORITMO GENÉTICO?	71
4. DESCRIPCIÓN DE LA HERRAMIENTA	73
4.1 INGRESO DE DATOS	73
4.1.1 Insumos	73
4.1.2 Actividades	74
4.1.3 Análisis de precios unitarios	75
4.1.4 Precedencias	76
4.1.5 Ingreso	76
4.1.6 Control	78
4.2 ALGORITMO GENÉTICO	80
4.2.1 Población	80
4.2.2 Población Intermedia	80
4.2.3 Población Mejores	80
4.2.4 Resultados Previos	81
4.2.5 Reporte	81
5. DETERMINACIÓN DE LOS PARÁMETROS DE FUNCIONAMIENTO DEL ALGORITMO GENÉTICO	82

5.1 DETERMINACIÓN DE LOS MEJORES OPERADORES PARA EL USO DE FCSP	84
5.1.1 Determinación del método de selección	84
5.1.2 Determinación del operador de recombinación	86
5.1.3 Determinación del operador de mutación	87
5.2 DETERMINACIÓN DE LOS MEJORES OPERADORES PARA EL USO DE MVPN	88
5.2.1 Determinación del método de selección	88
5.2.2 Determinación del operador de recombinación	90
5.2.3 Determinación del operador de mutación	92
6. ESTUDIO DE CASO	94
6.1 DESCRIPCIÓN DEL PROYECTO	94
6.2 ANÁLISIS DEL PROYECTO CON FCSP	100
6.2.1 Descripción de los parámetros del algoritmo genético empleados	100
6.2.2 Análisis de resultados	102
6.3 ANÁLISIS DEL PROYECTO CON MVPN	105
6.3.1 Descripción de los parámetros del algoritmo genético empleados	105
6.3.2 Análisis de resultados	106
6.4 COMPARACIÓN DE LOS RESULTADOS	109
7. CONCLUSIONES Y RECOMENDACIONES	112
8. ANOTACIONES SOBRE EL USO DE LA HERRAMIENTA BAJO EL AMBIENTE WINDOWS	114
BIBLIOGRAFÍA	115
ANEXOS	

LISTA DE CUADROS

	pág.
Cuadro 1. Planteamiento del ejemplo de la fábrica de pinturas	27
Cuadro 2. Valor de cada variable (x_1, x_2), valor de la utilidad z y valor de aptitud para cada solución en el ejemplo de la fábrica de pinturas.	28
Cuadro 3. Contiene la población, los valores de aptitud y la proporción de la circunferencia que corresponde a cada individuo	31
Cuadro 4. Valores de las variables (x_1, x_2), Aptitud, proporción y rango asociado a cada individuo	32
Cuadro 5. Individuos en cuyo rango quedó cada número aleatorio.	33
Cuadro 6. Contiene la población organizada de peor a mejor. La porción de circunferencia correspondiente a cada individuo se calcula a partir de los valores de aptitud asignados luego del ranking.	34
Cuadro 7. Contiene la población y la codificación en binarios de las variables x_1 y x_2 .	35
Cuadro 8. Contiene las actividades que componen el proyecto <i>Portal del Bosque</i> .	94
Cuadro 9. Contiene el plan de ventas del proyecto <i>Portal del Bosque</i>	96
Cuadro 10. Contiene los plazos de entrega del proyecto <i>Portal del Bosque</i>	97
Cuadro 11. Contiene la cantidad mínima de unidades a entregar por período para el proyecto <i>Portal del Bosque</i>	98
Cuadro 12. Contiene la cantidad de cuotas iniciales que se terminan de pagar en cada periodo para el proyecto <i>Portal del Bosque</i>	99
Cuadro 13. Contiene el resumen del presupuesto para el proyecto <i>Portal del Bosque</i>	100
Cuadro 14. Contiene la descripción de la población inicial en las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de FCSP	101
Cuadro 15. Contiene los parámetros del algoritmo genético empleados en las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de FCSP	102

Cuadro 16. Contiene el valor del flujo de caja acumulado mínimo obtenido con las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de FCSP	102
Cuadro 17. Contiene el ritmo de entregas obtenido con las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de FCSP	103
Cuadro 18. Contiene la descripción de la población inicial en las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de MVPN	105
Cuadro 19. Contiene los parámetros del algoritmo genético empleados en las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de MVPN	106
Cuadro 20. Contiene el valor presente neto de los flujos de caja netos, obtenido con las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de MVPN	106
Cuadro 21. Contiene el ritmo de entregas obtenido con las diferentes ejecuciones de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de MVPN	107
Cuadro 22. Contiene el ritmo de entregas obtenido con la ejecución A2_MVPN de la herramienta, para el proyecto <i>Portal del Bosque</i> con el uso de MVPN	108
Cuadro 23. Contiene el valor del flujo de caja acumulado mínimo obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto <i>Portal del Bosque</i>	109
Cuadro 24. Contiene el valor presente neto de los flujos de caja netos, obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto <i>Portal del Bosque</i>	109
Cuadro 25. Contiene el ritmo de entregas obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto <i>Portal del Bosque</i>	110

LISTA DE DIAGRAMAS

	pág.
Diagrama 1. Diagrama de flujo general de la función de evaluación	63
Diagrama 2. Diagrama de flujo de la función que calcula los días en que inicia y finaliza cada actividad	64
Diagrama 3. Diagrama de flujo de la función que calcula el costo directo de cada Actividad	65
Diagrama 4. Diagrama de flujo la función que calcula los egresos de dinero por período asociado a cada actividad y el egreso de dinero total por período	66
Diagrama 5. Diagrama de flujo de la función que calcula los ingresos de dinero asociados a las ventas	67
Diagrama 6. Diagrama de flujo de la función que calcula los ingresos de dinero asociados a las entregas y las penalizaciones por retrasos	68
Diagrama 7. Diagrama de flujo de la función que calcula el valor de aptitud para MVPN	69
Diagrama 8. Diagrama de flujo de la función que calcula el valor de aptitud para FCSP	70

LISTA DE GRÁFICAS

	pág.
Gráfica 1. Circunferencia dividida proporcionalmente según el valor de aptitud de cada individuo.	30
Gráfica 2. Longitud del Esquema vs. Probabilidad de Transmisión del Esquema, para un cromosoma de 100 genes.	56
Gráfica 3. Escalamiento del valor de aptitud.	58
Gráfica 4. Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de FCSP	84
Gráfica 5. Tiempos de ejecución del algoritmo con diferentes métodos de selección para el uso de FCSP	85
Gráfica 6. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de FCSP	86
Gráfica 7. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de FCSP	87
Gráfica 8. Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de MVPN	89
Gráfica 9. Proceso evolutivo obtenido con la combinación 2 (Torneo – Elitismo), para el uso de MVPN	90
Gráfica 10. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de MVPN	91
Gráfica 11. Proceso evolutivo obtenido con la combinación 8 (Torneo – Elitismo – 2P 2P), para el uso de MVPN	91
Gráfica 12. Proceso evolutivo obtenido con la combinación 7 (Torneo – Elitismo – 2P 1P), para el uso de MVPN	92
Gráfica 13. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de MVPN	93

LISTA DE FIGURAS

	pág.
Figura 1. Métodos de Búsqueda y Optimización	18
Figura 2. Diagrama de Flujo de un Algoritmo Genético	20
Figura 3. Codificación del Cromosoma (Kevin Petzold y Tarek Hegazy, 2003)	23
Figura 4. Codificación del Cromosoma (Sou-Sen Leu, Chung-Huei Yang y Jiun-Ching Huang, 1999)	24
Figura 5. Torneos jugados al azar para la selección de individuos en el ejemplo de la fábrica de pinturas.	29
Figura 6. Simulación plana de la ruleta rusa	32
Figura 7. Representación binaria de los individuos 4 y 6 del ejemplo de la Fábrica de Pinturas.	36
Figura 8. Recombinación con cruce en la tercera posición del cromosoma.	36
Figura 9. Dos cromosomas que comparten un esquema.	38
Figura 10. Codificación con alelos consecutivos	51
Figura 11. Codificación con alelos no consecutivos	52
Figura 12. Codificación circular con alelos consecutivos	53
Figura 13. Puntos de cruce disponibles en un cromosoma	55
Figura 14. Recombinación con un solo punto de cruce y dos padres	59
Figura 15. Recombinación con dos puntos de cruce y dos padres	60
Figura 16. Recombinación con dos puntos de cruce y tres padres	60
Figura 17. Herramienta, Hoja de Insumos	74
Figura 18. Herramienta, Hoja de Actividades	74
Figura 19. Herramienta, Hoja Unitarios	75
Figura 20. Herramienta, Hoja Precedencias	76
Figura 21. Herramienta, Hoja Ingreso	76
Figura 22. Herramienta, Hoja Ingreso	77
Figura 23. Herramienta, Hoja Control	79
Figura 24. Esquema de decisión para la comparación entre operadores genéticos	83

LISTA DE ANEXOS

	pág.
Anexo A. Estudio de caso: ejemplo controlado	120
Anexo B. Código de programación de la herramienta computacional	130

INTRODUCCIÓN

Los procesos de toma de decisiones en el sector de la construcción involucran un alto riesgo debido a las grandes sumas de dinero que se invierten en las obras civiles. Los constructores y en general los promotores de proyectos inmobiliarios buscan obtener un beneficio económico al desarrollar soluciones de vivienda y complejos comerciales. El inmenso universo de soluciones posibles y la falta de un soporte ágil y veraz para la toma de decisiones, no permite garantizar que el planteamiento del proyecto en la fase de planeación sea el que mejor resultados económicos ofrezca. Los algoritmos genéticos han sido propuestos y probados para la solución de problemas de este tipo en otros países, pero en Colombia su investigación y desarrollo es aún muy tímido. En este trabajo de grado se hace el desarrollo de una herramienta computacional basada en un algoritmo genético para proporcionar un soporte en la toma de decisiones en la fase de planeación de proyectos, con las consideraciones del caso colombiano.

Mediante la identificación de algunos factores de la realidad nacional y su influencia sobre el sector de la construcción, se logran identificar dos escenarios comunes bajo los cuales se plantean proyectos inmobiliarios en el país. Luego de la crisis económica de los años noventa ha sido poco usual que las empresas constructoras adquieran terrenos para desarrollar sus proyectos. Este cambio ha generado unas nuevas formas de negociación de los terrenos, lo cual tiene un impacto directo sobre la planeación de los proyectos. Este trabajo contempla dos escenarios enmarcados dentro de esta nueva realidad. El primero está condicionado por la capacidad de inversión o endeudamiento del constructor. En el segundo se busca obtener la mayor rentabilidad sobre la inversión.

La inclusión en la herramienta computacional de los dos escenarios mencionados, se hace con el desarrollo de dos funciones objetivo. Se desarrolla una función objetivo mediante la cual se busca encontrar la programación de obra con la cual se incurra en el menor nivel de

endeudamiento posible, y otra con la cual se busca maximizar el valor presente neto sobre la utilidad del proyecto. El resultado es una programación de obra ajustada a los requerimientos del proyectista.

La utilidad de la herramienta computacional y la aplicabilidad de los algoritmos genéticos en el caso colombiano, son comprobadas mediante el estudio de caso de un proyecto real. Así mismo se realiza un ejemplo controlado para comprobar la veracidad de los resultados obtenidos con el uso de la herramienta computacional. Se obtienen resultados magníficos al emplear los algoritmos genéticos como soporte para la programación de obras de construcción de edificaciones, y teniendo en cuenta las consideraciones que impone la realidad nacional.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar una herramienta computacional con la cual los atributos del proyecto evolucionen hasta encontrar una programación de obra que se ajuste al flujo de ingresos esperado.

OBJETIVOS ESPECÍFICOS

- Fortalecer y profundizar en los conceptos relacionados con algoritmos evolutivos aplicados a problemas comunes en el sector de la construcción.
- Identificar la codificación del cromosoma y los operadores de reproducción que mejor se ajusten a este tipo de problemas.
- Obtener el costo, duración, fecha de iniciación y fecha de finalización de cada actividad, capítulo y de la totalidad de la obra, ajustado al momento financiero del proyecto según el flujo de fondos.

1. REVISIÓN BIBLIOGRÁFICA

1.9 ¿QUÉ SON LOS ALGORITMOS GENÉTICOS?

El investigador de la Universidad de Michigan, John Holland desarrolló una técnica a la cual se le llamó originalmente "planes reproductivos", sin embargo, sólo se hizo popular luego de 1975 como algoritmos genéticos¹(AGs). Cuando Holland desarrolló los algoritmos genéticos, lo hizo con un propósito diferente al que se les imputa hoy en día. Él quería lograr dos objetivos, el primero era abstraer y explicar el proceso de adaptación de la naturaleza y el segundo era diseñar un sistema artificial ó programa informático que conservara los mecanismos de la selección natural². El propósito inicial de Holland estaba más ligado a la biología que a cualquier otra ciencia.

Aunque la motivación principal de Holland residía en la biología, una de las utilidades más populares de los algoritmos genéticos se encuentra en la investigación de operaciones. Los algoritmos genéticos no incursionaron en dicha disciplina de forma inmediata, según Dowsland³ porque el desarrollo original de los AGs había sido llevado a cabo por un grupo muy selecto o círculo reducido de investigadores de la Universidad de Michigan. Luego, el número de investigadores en este tema se incrementó y se formalizaron cinco conferencias sobre el tema en Estados Unidos y tres en Europa. Las publicaciones sobre computación evolutiva (otra denominación para los AGs) sólo eran tenidas en cuenta por editoriales concentradas en la inteligencia artificial, lo que provocó que las investigaciones sobre algoritmos genéticos no fueran de fácil dominio para quienes trabajaban en la investigación de operaciones.

¹ COELLO COELLO, Carlos A. Revista Red Científica [online]. [Citado: Septiembre 10 de 2004]. Disponible en <http://www.redcientifica.com/doc/doc199904260011.html>

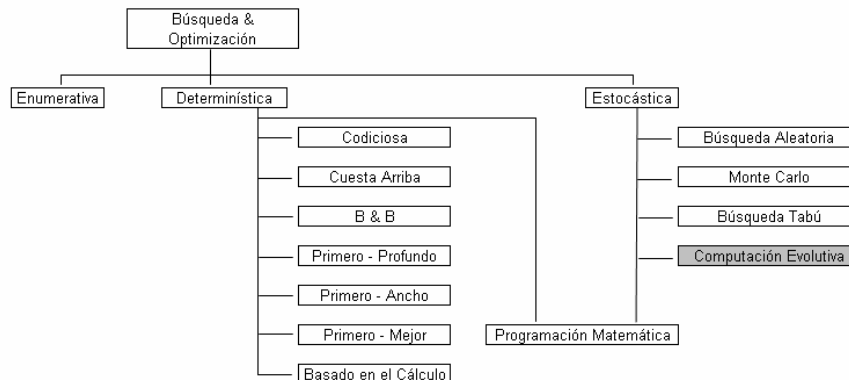
² GOLDBERG E., David. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989. p.1

³ DOWSLAND, Kathryn A. Genetic Algorithms-A Tool for OR? [online]. The Journal of the Operational Research Society, Vol. 47 No. 4. Apr., 1996. [Citado Diciembre 1 de 2003] Disponible en: <http://links.jstor.org/sici?sici=0160-5682%28199604%2947%3A4%3C550%3AGATFO%3E2.0.CO%3B2-1>

El paso de los años hizo que la información se filtrara hacia la investigación de operaciones. Uno de los primeros problemas en ser abordado fue el conocido como: *el agente viajero*, el experimento funcionó, pero existía muy poca información práctica sobre el uso de los algoritmos genéticos en la investigación de operaciones. Se generó entonces un gran interés en investigar la comparación entre los métodos clásicos de optimización y la optimización con algoritmos genéticos. A partir de ese auge, surgieron una gran cantidad de investigaciones en cuanto a la forma de abordar las restricciones con esta nueva técnica y se crearon aplicaciones de los algoritmos genéticos para la industria y el mundo de los negocios⁴.

El siguiente diagrama ilustra el lugar donde se ubican los algoritmos genéticos dentro de los métodos de búsqueda y optimización:

Figura 1. Métodos de Búsqueda y Optimización⁵



Los algoritmos genéticos constituyen una herramienta poderosa al momento de buscar soluciones para un problema en función de sus características. Éstos funcionan con base en la teoría de la evolución de Darwin, que sostiene que sólo el individuo más apto logra

⁴ Ibid.,

⁵ COELLO COELLO, Carlos A.; VAN VELDHUIZEN, David A. y LAMONT, Gary B. Evolutionary Algorithms for Solving Multi-Objective Problems. Plenum Publishers. P. 19.

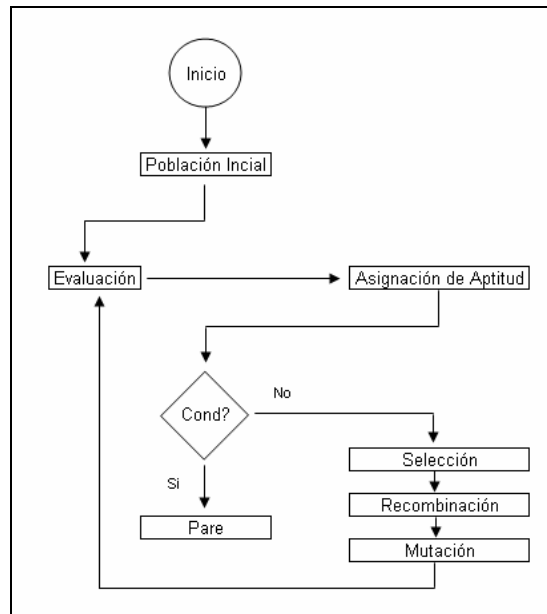
sobrevivir. Emulan la teoría de la evolución de las especies pues a partir de varias soluciones a un problema se logra que la solución más apta sea la que finalmente sobreviva.

Las soluciones al problema se codifican en un arreglo de caracteres. A éste se le llama cromosoma y contiene toda la información que permite dar solución al problema. Por ejemplo si se va a maximizar $f(x) = \text{sen}(x)$, entonces el cromosoma contiene la información con la cual se construye la variable x . Dicha variable puede estar codificada en forma binaria o en parámetros reales, es decir que en el cromosoma está codificado el número como tal o puede estar la representación binaria del número.

Al igual que los cromosomas están compuestos por genes, los problemas están compuestos por variables que contienen información de su naturaleza. Los algoritmos genéticos tratan de modelar soluciones para los problemas basándose en sus características, como las restricciones y variables que lo componen. Para un problema existen diferentes soluciones acertadas pero también existen alternativas mejores o ajustadas a un resultado esperado. De esta forma, si se parte de dos soluciones acertadas y se mezclan y conjugan entre sí - analizando al mismo tiempo los costos y beneficios de cada característica- es posible lograr un resultado de solución mejor a las dos planteadas inicialmente. Así mismo, los cromosomas se mezclan y comparten la información genética para formar nuevas generaciones cada vez mejores en virtud de las buenas características heredadas de generaciones predecesoras. Las restricciones planteadas en el problema y ciertas penalizaciones que se imparten a los cromosomas que no se adaptan a ellas, constituyen los parámetros con los cuales se mide su fortaleza. Cada cruce de dos cromosomas o mutación de un solo cromosoma produce un nuevo cromosoma de nueva generación.

Un diagrama general de flujo de un algoritmo genético es el siguiente:

Figura 2. Diagrama de Flujo de un Algoritmo Genético⁶



A partir de una población inicial de cromosomas, se procede a evaluar la bondad de cada individuo, luego se asigna un valor asociado a esa bondad, mediante una función de aptitud. Consecutivamente, se evalúa si la alternativa es suficientemente buena para dar una solución definitiva al problema; de ser así, el proceso iterativo se detiene, en caso contrario deberá continuar. Posteriormente ocurre el proceso de reproducción, compuesto por la selección, la recombinación y la mutación. En el proceso de selección, se escogen los mejores individuos según el valor de aptitud que se les ha asignado, luego en la recombinación se intercambia la información genética de unos y otros y finalmente en algunos casos se emplea la mutación, que consiste en cambiar de alguna forma la información de uno o más genes dentro del cromosoma.

Se dice entonces que los algoritmos genéticos son un método de búsqueda altamente paralelo⁷ que basa su funcionamiento en la selección de individuos que sean capaces de

⁶ KALYANMOY, Deb. Multi-Objective Optimization using Evolutionary Algorithms. Willey, 2001. p.87.

⁷ SANTO ORCERO, David. ¿Que es un Algoritmo Genético? [online]. [Citado: Septiembre 10 de 2004]. Disponible en: <http://www.orcero.org/irbis/disertacion/node189.html>. p. 3.

representar una solución a un problema adaptándose a las restricciones propias del mismo.

1.10 INTERCAMBIO TIEMPO-COSTO

Para el caso específico de este trabajo de grado, se considera que un proyecto de construcción está compuesto por actividades cada una de las cuales requiere recursos como insumos y mano de obra. El costo de la actividad no es fijo, pues depende del tiempo de ejecución destinado para realizarla. Cuando se requiere “acelerar la marcha” es necesario emplear más mano de obra en la ejecución del proyecto y como consecuencia, el costo de las actividades aumenta y el tiempo de ejecución disminuye.

Por lo tanto el intercambio tiempo-costo consiste en el análisis sobre el impacto en un proyecto cuando se comparan diferentes costos y duraciones de las actividades. Encontrar una relación tiempo-costo que satisfaga el escenario del proyectista requiere de una gran cantidad de cálculos debido a la gran cantidad de permutaciones⁸.

En un proyecto de 50 actividades, cada una con dos métodos de construcción diferentes se tendrían $2^{50} (= 1,13 \times 10^{15})$ opciones diferentes para realizarlo. La cifra automáticamente descarta la posibilidad de analizar el intercambio tiempo costo de forma exhaustiva, porque un computador que permita examinar un millón de opciones por segundo tardaría 35.7 años en examinarlas.

El intercambio tiempo costo es muy importante, pero trae consigo grandes retos al momento de analizar la información. La gran cantidad de permutaciones existentes en los problemas de esta naturaleza hacen que las herramientas como los algoritmos genéticos sean de gran utilidad para estos casos.

⁸ HEGAZY, Tarek. Computer-Based Construction Project Management. Prentice Hall 2002. p. 212

1.11 LA CODIFICACIÓN DEL CROMOSOMA

La codificación del cromosoma tiene un gran impacto en el funcionamiento del algoritmo genético. La optimización es función de la representación de los datos⁹, una codificación errada puede conducir incluso a la necesidad de abortarla pues las soluciones obtenidas pueden no cumplir con las restricciones del problema. Por el contrario, si se escoge acertadamente puede conducir a una convergencia rápida y además facilita la programación del algoritmo.

Entre los tipos (binario y parámetros reales) de codificación no existe uno absolutamente mejor que el otro. Cuando se emplea la codificación con parámetros reales, ocurre un acercamiento entre la optimización clásica y los algoritmos genéticos, pero no existe una gran cantidad de estudios sobre esta forma de cromosomas¹⁰. Se hace evidente que para escoger la forma de codificación del cromosoma es necesario revisar los trabajos anteriores sobre temas acordes con el problema al que se desea dar solución.

Para efectos de este trabajo de grado, se revisaron trabajos de nivelación de inventarios, asignación de recursos, programación de obra e intercambio tiempo-costo (time cost trade-off, TCT por su sigla en inglés). Se consultó el tipo de codificación empleado por diferentes autores, y la forma como organizaron el cromosoma.

Kevin Petzold y Tarek Hegazy¹¹ en su trabajo “Genetic Optimization for Dynamic Project Control” presentaron un modelo práctico para determinar acciones correctivas óptimas durante la ejecución de proyectos de construcción. El modelo propuesto se diseñó de forma tal que la información almacenada sobre los recursos se emplee para estimar la programación, y el control de las actividades restantes de forma dinámica, es decir paralelo a la construcción de la obra. Las actividades del proyecto están predefinidas y cada una

⁹ SANTO ORCERO, David. Op. Cit. p. 3.

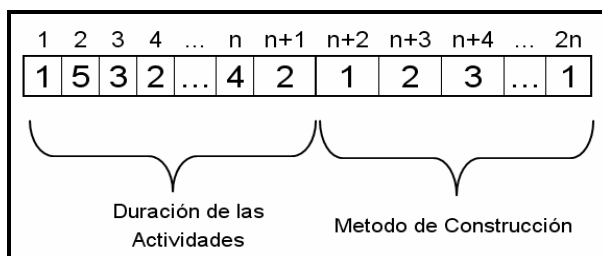
¹⁰ KALYANMOY, Deb. Op. Cit., p. 124.

¹¹ HEGAZY, Tarek and PETZOLD, Kevin. Genetic Optimization for Dynamic Project Control. Journal of Construction Engineering and Management. ASCE, July/August 2003.

tiene información asociada; tres métodos de construcción para cada actividad, con base en esa información se calcula a partir del rendimiento y tiempo de duración de la actividad, el costo en que se ha incurrido. De esa forma si se quiere obtener información en un momento dado y hay alguna actividad incompleta en ese momento, entonces el costo y tiempo restante de la misma son proporcionales a la duración de la actividad.

En ese trabajo, la codificación del cromosoma se hizo con parámetros reales. El cromosoma se construyó de la siguiente forma:

Figura 3. Codificación del Cromosoma (Kevin Petzold y Tarek Hegazy, 2003)



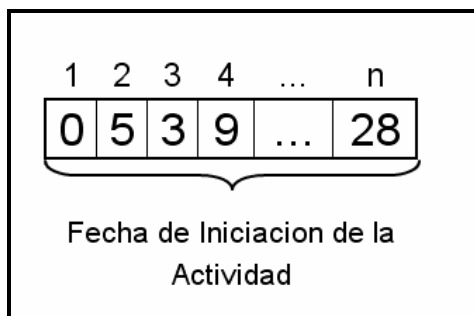
En el trabajo sobre nivelación de inventarios en construcción realizado por Sou-Sen Leu, Chung-Huei Yang y Jiun-Ching Huang¹², también se emplea la codificación con parámetros reales. Estos autores desarrollaron una herramienta basada en algoritmos genéticos que produce una programación de obra para cumplir con las metas de tiempo y costos y utilizaron una función objetivo para nivelar los inventarios. La nivelación de inventarios no es otra cosa que homogenizar el gasto de insumos en todos los periodos del proyecto. La función objetivo definida consiste en minimizar la diferencia entre el gasto de material de cada periodo y la media del gasto de material para todos los periodos.

Los autores supusieron una duración constante de las actividades, el gasto de material constante a lo largo de la duración de la actividad, y la relación de precedencia entre

¹² LEU, Sou-Sen; YANG, Chung-Huei y HUANG, Jiun-Ching. Resource Leveling in Construction by Genetic Algorithm-based Optimization and its Decision Support System Application. Department of Construction Engineering, National Taiwan University of Science and Technology. April, 1999. Disponible en: <http://www.elsevier.com/locate/autocon>

actividades fue fijada previamente. La codificación del cromosoma empleada en dicho trabajo es la siguiente:

Figura 4. Codificación del Cromosoma (Sou-Sen Leu, Chung-Huei Yang y Jiun-Ching Huang, 1999)



Otro trabajo similar es el de Sou-Sen Leu con Chung-Huei Yang¹³. Es muy parecido en su planteamiento al anterior. Lo que buscan encontrar con la herramienta que describen en el artículo es la combinación óptima de duración de las actividades que provea una nivelación de los inventarios y el costo y duración mínimos del proyecto. La codificación empleada es la misma, pero ahora la variable de decisión representada en el cromosoma es la duración de la actividad. Este cambio en la variable de decisión se debe a que incluyen TCT, y ello implica que la duración de las actividades sea variable y no fija como en el trabajo anteriormente expuesto. El TCT busca un balanceo entre costos y duración del proyecto.

En el trabajo realizado por Y. Cengiz Toklu¹⁴, se emplea una codificación del cromosoma similar a las expuestas. El profesor Toklu realizó la codificación del cromosoma con parámetros reales. El cromosoma contiene la información sobre la fecha de inicio de cada actividad. Toklu concluyó de su trabajo que la programación de obra realizada con el método de la ruta crítica no es fácil de implementar para proyectos con restricciones en los

¹³ LEU, Sou-Sen and YANG, Chung-Huei. GA-Based Multicriteria Optimal Model for Construction Scheduling. Journal of Construction Engineering and Management. ASCE. November/December, 1999.

¹⁴ TOKLU Y., Cengiz. Application of genetic Algorithms to Construction Scheduling with or without resource Constraints. Canadian Journal of Civil Engineering. June 2002.

recursos, mientras que la programación con algoritmos genéticos es igualmente fácil con o sin restricciones.

La codificación del cromosoma más empleada cuando se trata de afrontar problemas que incluyan programación de obra, TCT, asignación de recursos y nivelación de inventarios, es la codificación con parámetros reales. La información que provee el cromosoma con parámetros reales puede ser inmediatamente empleada en la función objetivo y en las funciones que asignan valores de aptitud al cromosoma. No tendría sentido emplear la codificación binaria ya que la información en el cromosoma no estaría representando las variables de decisión como tal, además la codificación binaria puede tener ciertas desventajas. Darrel Whitley¹⁵ expone el siguiente ejemplo: el número 15 y el número 16, son consecutivos en su representación decimal, aunque no en su representación binaria (15=01111, 16=10000), es decir no comparten ningún bit. Es mejor para efectos de la búsqueda que los números adyacentes sean diferentes por un solo bit. También sostiene Whitley que la representación natural del problema debe ser igual a la representación en el cromosoma.

1.12 EVALUACIÓN Y APTITUD

La categorización de los cromosomas se hace a partir de su aptitud hacia el problema. Cuando se tiene una población muy grande de cromosomas o individuos, es necesario saber qué tan bueno es cada uno de ellos para proceder a seleccionar su información y transmitirla a nuevas generaciones. Para ello se asigna una función de aptitud y se evalúa para cada individuo. La función arroja un valor numérico, el cual permite conocer qué tan bueno es cada uno de los individuos.

La “bondad” de cada cromosoma se evalúa a partir de la función objetivo y las restricciones, con una función de aptitud. Cuando los problemas tienen restricciones es

¹⁵ WHITLEY, Darrel. Genetic Algorithms and Evolutionary Computing [online]. Computer Science Department, Colorado State University. [Citado Octubre 6 de 2004]. Disponible a través de: whitley@cs.colostate.edu

necesario implementar penalizaciones. Goldberg¹⁶ indica que las restricciones deben plasmarse en forma de inecuaciones y el incumplimiento de las restricciones debe penalizarse en la función de aptitud. Es decir que el incumplimiento de una o alguna de las restricciones le resta aptitud al individuo y por ende se hace menos probable que su información sea transmitida.

Es necesario hacer claridad en la diferencia que existe entre la función objetivo y la función de aptitud. La función objetivo sólo define la ecuación que se busca optimizar, mientras que la función de aptitud tiene en cuenta el valor de la función de objetivo y también el cumplimiento o incumplimiento de las restricciones¹⁷. Los individuos se seleccionan según su aptitud, a partir de dicho valor se hace la categorización sobre cuáles cromosomas, y en qué proporción, van a contribuir con su información en las nuevas generaciones. La selección de los individuos tiene que estar entonces precedida por la asignación de aptitud.

1.13 TÉCNICAS DE SELECCIÓN

La selección de los cromosomas que van a transmitir información genética a las siguientes generaciones se hace con base en su aptitud. El primordial objetivo de esta operación es escoger las buenas soluciones y eliminar las malas. Existen varios tipos de selección entre los cuales se encuentran: la selección por torneo, la selección proporcional, y la selección por ranking.

La forma más clara de explicar como opera cada técnica de selección es mediante un ejemplo. Se va a desarrollar el siguiente ejemplo sobre una fábrica de pinturas, y a través de éste, se hará la explicación sobre cada una de las técnicas de selección:

Una empresa produce pinturas para interiores y para exteriores para lo cual emplea dos materias primas M1 y M2 de la siguiente forma:

¹⁶ GOLDBERG E., David. Op. cit. p. 85.

¹⁷ COELLO COELLO, Carlos A.; VAN VELDHUIZEN, David A. y LAMONT, Gary B. Op. cit., p. 23.

Cuadro 1. Planteamiento del ejemplo de la fábrica de pinturas¹⁸.

	Toneladas de materia prima por tonelada de producto terminado		Disponibilidad máxima diaria de materia prima en toneladas
	Pintura Exterior	Pintura Interior	
M1	6	4	24
M2	1	2	6
Utilidad por tonelada de producto terminado	5	4	

La empresa quiere determinar la cantidad de pintura para interiores y exteriores que debe producir diariamente para maximizar su utilidad; teniendo en cuenta que la demanda máxima de pintura para interiores es de dos toneladas y que la demanda diaria de pintura para interiores no puede exceder a la demanda de pintura para exteriores por más de una tonelada.

Variables:

x_1 = Producción diaria de pintura para exteriores en toneladas

x_2 = Producción diaria de pintura para interiores en toneladas

Modelo:

$$\text{Maximizar } 5x_1 + 4x_2 = z$$

Sujeto A:

$$6x_1 + 4x_2 \leq 24 \quad (1)$$

¹⁸ Ejemplo adaptado del problema de la Wyndor Glass Co. Citado en: LIEBERMAN y HILLIER. Investigación de Operaciones. México: McGraw Hill, 2002. p. 27.

$$1x_1 + 2x_2 \leq 6 \quad (2)$$

$$x_2 \leq 2 \quad (3)$$

$$x_2 \geq x_1 + 1 \quad (4)$$

$$x_1, x_2 \geq 0 \quad (5)$$

Se tiene la siguiente población de 6 individuos:

Cuadro 2. Valor de cada variable (x_1, x_2), valor de la utilidad z y valor de aptitud para cada solución en el ejemplo de la fábrica de pinturas.

Solución #	X1	X2	Z	Aptitud	Penalidad 1	Penalidad 2	Penalidad 3	Penalidad 4	Penalidad 5
1	1,0	2,0	13,0	13,0	0	0	0	0	0
2	0,8	2,0	12,0	2,0	0	0	0	-10	0
3	0,7	1,5	9,5	9,5	0	0	0	0	0
4	2,5	1,5	18,5	18,5	0	0	0	0	0
5	4,0	2,0	28,0	8,0	-10	-10	0	0	0
6	1,4	1,8	14,2	14,2	0	0	0	0	0

El cromosoma es cada par de datos x_1 y x_2 , entonces x_1 y x_2 son dos genes que componen un cromosoma. La utilidad z , es el valor a maximizar, y se construye a partir de la información que contiene cada individuo o cromosoma. Las penalidades consisten en restar 10 unidades al valor z por el incumplimiento de alguna restricción; la penalidad 1 corresponde a la restricción 1 y así hasta la penalidad 5. La columna z de la Cuadro 2, muestra el valor que se busca maximizar mientras que la columna aptitud muestra el valor z menos el valor de las penalidades por incumplimiento de las restricciones.

Se observa con claridad que la solución (3) es mejor que la solución (2); aunque la utilidad para (2) sea 12 unidades y la utilidad para (3) sea 9,5 unidades. Esto ocurre debido a que la solución (2) incumple una de las restricciones entonces se penaliza su utilidad. De esta forma, la solución (3) tiene más probabilidad de ser seleccionada que la solución (2). Las técnicas de selección operan basándose en la categorización de las soluciones según el

valor de aptitud asignado a cada individuo. El ejemplo del cuadro 1 sirve como herramienta para desarrollar las técnicas de selección que se van a explicar en esta sección.

1.13.1 Selección por torneo. Este tipo de selección opera haciendo torneos entre pares de soluciones y haciendo copia de la solución que gana cada torneo. Con las copias de los individuos ganadores se conforma una nueva población. Se deben realizar tantos torneos como individuos se desean obtener luego de la selección.

Para efectos del ejemplo se realizan torneos entre las parejas 1-2, 3-4, 5-6, 1-6, 2-4, y 3-5. El resultado es el siguiente:

Figura 5. Torneos jugados al azar para la selección de individuos en el ejemplo de la fábrica de pinturas.

	Solución #	Aptitud	Seleccionados
Torneo 1	1	13,0	1
	vs.		
	2	2,0	
Torneo 2	3	9,5	4
	vs.		
	4	18,5	
Torneo 3	5	8,0	6
	vs.		
	6	14,2	
Torneo 4	1	13,0	6
	vs.		
	6	14,2	
Torneo 5	2	2,0	4
	vs.		
	4	18,5	
Torneo 6	3	9,5	3
	vs.		
	5	8,0	

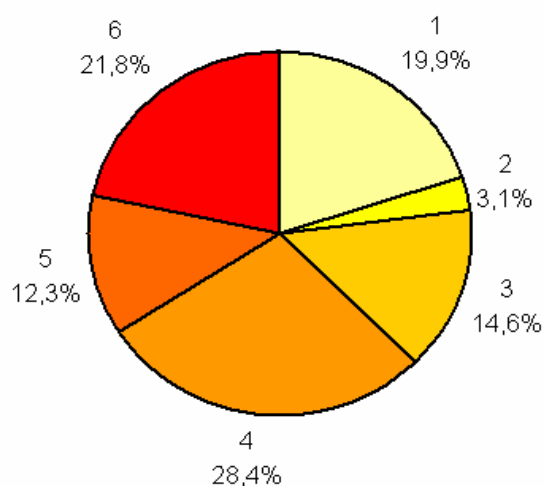
Es así como se forma la nueva población, que contiene a los individuos que mejor solucionan el problema. Se observa por ejemplo que los individuos (4) y (6), los cuales tienen asignada la mayor aptitud en la población original, están presentes dos veces cada

uno en la nueva población. La solución (2) y la solución (5) no fueron tenidas en cuenta para conformar la nueva población debido a que tienen el menor valor de aptitud dentro de la población original.

La selección por torneo opera mediante la comparación entre pares de individuos. De esa forma se espera que los individuos más aptos sean seleccionados y los menos aptos sean descartados.

1.13.2 Selección proporcional. A este tipo de selección se le puede asociar con una ruleta ya que funciona bajo el mismo principio. Se conforma una circunferencia y a cada individuo de le asigna una porción proporcional a la magnitud de su valor de aptitud¹⁹. Si dicha ruleta se pone a girar y existe un indicador fijo, hay más probabilidad de que el indicador apunte a la porción de mayor magnitud cuando ésta se detenga. Para el ejemplo que se viene desarrollando resultaría la siguiente ruleta:

Gráfica 1. Circunferencia dividida proporcionalmente según el valor de aptitud de cada individuo.



¹⁹ GOLDBERG E., David. Op. cit., p.11

Que proviene del siguiente cuadro:

Cuadro 3. Contiene la población, los valores de aptitud y la proporción de la circunferencia que corresponde a cada individuo

Solución #	X1	X2	$f(x)$	$\frac{f(x)}{\sum f(x)}$
			Aptitud	Proporción
1	1,0	2,0	13,0	0,199
2	0,8	2,0	2,0	0,031
3	0,7	1,5	9,5	0,146
4	2,5	1,5	18,5	0,284
5	4,0	2,0	8,0	0,123
6	1,4	1,8	14,2	0,218
Suma			65,2	1,00
Promedio			10,9	0,17

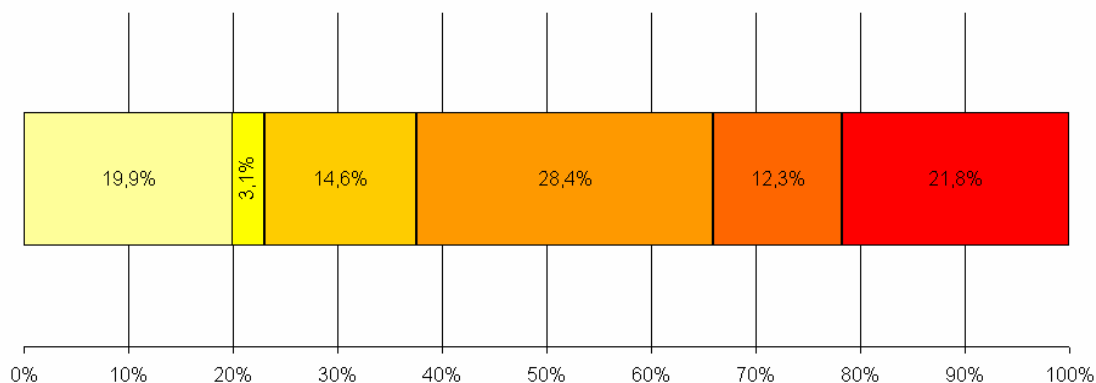
En la Gráfica 1 se observa que el individuo (4), tiene asociado el mayor valor de aptitud, entonces tiene la mayor porción de la circunferencia, mientras que la solución (2) tiene la menor porción de la circunferencia ya que su valor de aptitud es el menor de todos.

El profesor Kalyanmoy plasma en su obra²⁰ una forma de simular la ruleta en forma plana. La suma de las porciones de circunferencia que le corresponde a cada individuo es igual uno. A cada individuo se le asigna un rango de la misma magnitud que la porción de circunferencia que le corresponde. Luego se organizan dichos rangos de forma que el límite inferior del primero sea 0 y su límite superior sea el valor de la proporción de circunferencia que le corresponde; el segundo rango tiene como límite inferior el valor de la proporción de circunferencia que corresponde al individuo anterior y como límite superior el valor de la suma de su límite inferior más el valor de la proporción de circunferencia que a él le corresponde. Así sucesivamente hasta el último individuo cuyo rango tendrá como

²⁰ KALYANMOY, Deb. Op. cit., p. 91.

límite superior el número 1. Luego se generan tantos números aleatorios como individuos haya en la población, números aleatorios entre 1 y 0. Cada número generado está contenido en el rango de algún individuo. Cuando el número aleatorio cae dentro del rango de alguno de los individuos entonces dicho individuo será seleccionado. El individuo cuyo valor de aptitud sea mayor, tendrá un rango más grande y por eso la probabilidad de ser seleccionado es mayor. En el ejemplo que se viene desarrollando se vería de la siguiente forma:

Figura 6. Simulación plana de la ruleta rusa



La barra mostrada en la Figura 6 contiene los rangos correspondientes a todos los individuos organizados consecutivamente. La construcción de esta barra proviene de los valores presentados en el cuadro 4.

Cuadro 4. Valores de las variables (x_1, x_2), Aptitud, proporción y rango asociado a cada individuo

$$f(x) = \frac{f(x)}{\sum f(x)}$$

Solución #	X1	X2	Aptitud	Proporción	P acum.
1	1,0	2,0	13,0	0,199	0,199
2	0,8	2,0	2,0	0,031	0,230
3	0,7	1,5	9,5	0,146	0,376
4	2,5	1,5	18,5	0,284	0,660
5	4,0	2,0	8,0	0,123	0,782
6	1,4	1,8	14,2	0,218	1,000
Suma			65,2	1,00	
Promedio			10,9	0,17	

En la casilla P acum. se observan los límites superiores del rango de cada individuo. Por ejemplo el rango del individuo (2) es (0,199 , 0,230], mientras que el rango de la solución (4) es (0,376 , 0,66]. Hay más probabilidad de que un número aleatorio tenga un valor dentro del rango de (4) que dentro del rango de (2) ya que el rango de (4) es más grande.

Para proseguir con el ejemplo de la fábrica de pinturas, se crean seis números aleatorios: 0,173, 0,596, 0,981, 0,477, 0,62, 0,843 usando un distribución uniforme [0,1]. Cada uno de ellos está contenido dentro del rango de los siguientes individuos presentados en el cuadro 5.

Cuadro 5. Individuos en cuyo rango quedó cada número aleatorio.

Num. Aleatorio	En el rango de:
0,173	1
0,596	4
0,981	6
0,477	4
0,620	4
0,843	6

El individuo (4) fue seleccionado tres veces, el individuo (6) fue seleccionado dos veces y el individuo (1) fue seleccionado una vez. Los individuos (2), (3) y (5) fueron descartados.

La técnica de selección proporcional es más exigente al momento de programarse debido a que es necesario simular todo el proceso de la ruleta expuesto en esta sección. Además considera el profesor Kalyanmoy²¹ que este tipo de selección presenta problemas cuando los valores de aptitud entre los individuos son o muy parejos o muy dispares. Cuando éstos son muy parejos, o mejor, cuando la proporción de circunferencia que les corresponde es muy similar para todos, entonces es indiferente hacer o no hacer el proceso de selección ya que la probabilidad de ser escogido para cualquier individuo es muy similar a la de los demás. Cuando ocurre lo contrario puede presentarse una solución cuyo valor de aptitud le

²¹ KALYANMOY, Deb. Op. cit., p. 92.

permita apoderarse de una proporción mayor de la circunferencia. El problema con ello sería que tal individuo sería seleccionado muchas veces y se descartarían otras soluciones que si bien no son tan buenas, mantienen la diversidad dentro de la población.

1.13.3 Selección por ranking. Éste tipo de selección opera mediante la organización de los individuos según su valor de aptitud. Se enumeran en una lista de peor a mejor y se les asigna un nuevo valor de aptitud igual al puesto en el cual quedaron²². Finalmente se hace el proceso de selección proporcional sobre los individuos a partir del nuevo valor de aptitud.

Cuadro 6. Contiene la población organizada de peor a mejor. La porción de circunferencia correspondiente a cada individuo se calcula a partir de los valores de aptitud asignados luego del ranking.

Solución #	X1	X2	Aptitud	Aptitud 2	Proporción	P. acum.
2	0,8	2,0	<u>2,0</u>	1,0	0,048	0,048
5	4,0	2,0	<u>8,0</u>	2,0	0,095	0,143
3	0,7	1,5	<u>9,5</u>	3,0	0,143	0,286
1	1,0	2,0	<u>13,0</u>	4,0	0,190	0,476
6	1,4	1,8	<u>14,2</u>	5,0	0,238	0,714
4	2,5	1,5	<u>18,5</u>	6,0	0,286	1,000
Suma			<u>65,2</u>	21,0	1,000	
Promedio			16,3	3,5	0,2	

En el cuadro 6 se observa que al optar por los mismos números aleatorios del numeral anterior entonces: el individuo (6) será seleccionado tres veces, el individuo (4) será seleccionado dos veces, el individuo (3) será seleccionado una vez y los individuos (1), (2) y (5) serán descartados.

1.13.4 Elitismo. Este operador se emplea para preservar por fuerza las mejores soluciones. Kalyanmoy²³ declara la existencia de muchas implementaciones del elitismo,

²² KALYANMOY, Deb. Op. cit., p. 92.

²³ KALYANMOY, Deb. Op. cit., p.107.

pero explica que éste funciona de forma general haciendo una copia a la siguiente generación del x porcentaje de individuos con los valores de aptitud más altos.

1.14 LA RECOMBINACIÓN

La reproducción se ha iniciado ya con el proceso de selección. Tal proceso es anterior a la recombinación pues se busca que a ésta instancia lleguen sólo los individuos –dentro de la población- que mejor información genética contengan. En la recombinación se mezcla la información genética de los individuos seleccionados para generar nuevos individuos presumiblemente mejores.

Para explicar bien cómo opera la recombinación, se continúa con el ejemplo de la fábrica de pinturas. Cada individuo está compuesto de un par de variables (x_1 y x_2), con las cuales se da solución a la función de utilidad y validez a cada una de las restricciones. Si el cromosoma estuviera codificado de forma binaria se tendría lo siguiente:

Cuadro 7. Contiene la población y la codificación en binarios de las variables x_1 y x_2 .

Solución #	X1	X2	X1	X1
1	1,0	2,0	001010	010100
2	0,8	2,0	001000	010100
3	0,7	1,5	000111	001111
4	2,5	1,5	011001	001111
5	4,0	2,0	101000	010100
6	1,4	1,8	001110	010010

Los cromosomas de los individuos (4) y (6) serían los siguientes:

Figura 7. Representación binaria de los individuos 4 y 6 del ejemplo de la Fábrica de Pinturas.

Solución #		Cromosoma					
		X1	X2	X3	X4	X5	X6
4	X1	0	1	1	0	0	1
	X2	0	0	1	1	1	1
6	X1	0	0	1	1	1	0
	X2	0	1	0	0	1	0

La recombinación consiste en intercambiar la información contenida en los cromosomas escogiendo uno o más puntos de cruce. Por ejemplo (4) está compuesto por una serie de 6 alelos: $\{a_1, a_2, \dots, a_6\}$ en donde: $a_1 = \{0,0\}$, $a_2 = \{1,0\}$, $a_3 = \{1,1\}$, $a_4 = \{0,1\}$, $a_5 = \{0,1\}$ y $a_6 = \{1,1\}$; El cromosoma (6) está compuesto también por una serie de 6 alelos: $\{b_1, b_2, \dots, b_6\}$. Si se determina el punto de cruce a partir de la tercera posición del cromosoma, resultaría un hijo cuyo cromosoma también estaría compuesto por 6 alelos pero la serie sería la siguiente: $\{a_1, a_2, a_3, b_4, b_5, b_6\}$. Gráficamente se observaría así:

Figura 8. Recombinación con cruce en la tercera posición del cromosoma.

Solución #		Cromosoma						Aptitud
		X1	X2	X3	X4	X5	X6	
4	X1	0	1	1	0	0	1	18,5
	X2	0	0	1	1	1	1	
6	X1	0	0	1	1	1	0	14,2
	X2	0	1	0	0	1	0	
Hijo	X1	0	1	1	1	1	0	19,0
	X2	0	0	1	0	1	0	

La recombinación trata de recrear la reproducción hembra-macho, en la cual la información del padre se mezcla con la de la madre para producir un nuevo individuo. El hijo, ése nuevo individuo que resulta de la recombinación, puede ser mejor que sus padres pues podría heredar lo mejor del uno y del otro. En el ejemplo se observa con claridad que el hijo que resulta de la recombinación entre (4) y (6) tiene un valor de aptitud de 19 unidades, valor superior al de sus dos padres. Sin embargo, no siempre ocurre de esa forma, puede ser que el hijo de dos muy buenos individuos resulte menos apto que sus padres²⁴. Eso no trae consigo mayores problemas pues en la siguiente iteración, en el proceso de selección, dicho individuo será descartado.

1.15 LA MUTACIÓN

La mutación es el operador genético más sencillo de todos pero tiene una gran ingerencia en el funcionamiento de un AG. Consiste en cambiar uno o más genes dentro del cromosoma de forma aleatoria o según la regla que imponga el creador del algoritmo. La mutación tiene la función de recobrar información valiosa que puede perderse en la recombinación y la selección²⁵.

1.16 LOS ESQUEMAS, ¿POR QUÉ FUNCIONAN LOS AGS?

Existe una explicación fundamentada matemáticamente para corroborar como, a partir de cromosomas, es posible lograr un método de búsqueda robusto; dicha explicación encuentra fundamento en la teoría de probabilidades.

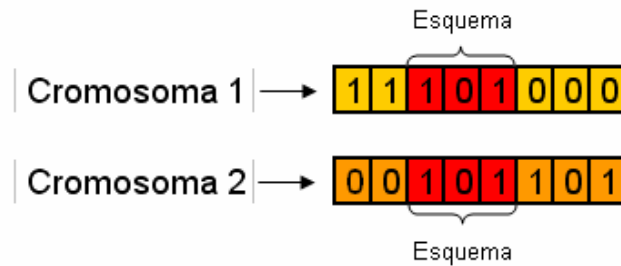
Antes de comenzar con la explicación formal sobre cómo usan los AGs la probabilidad para direccionar su búsqueda, es necesario definir y explicar lo que es un esquema. Dentro de la cadena de genes que componen un cromosoma puede existir una sub cadena que

²⁴ KALYANMOY, Deb. Op. cit., p. 94.

²⁵ GOLDBERG E., David. Op. cit., p.14

contiene cierta información en una determinada posición dentro del cromosoma. Gráficamente se observa de la siguiente forma:

Figura 9. Dos cromosomas que comparten un esquema.



En la Figura 9 ambos cromosomas comparten un esquema, en este caso (1,0,1), desde la posición 3 hasta la posición 5.

Los esquemas son una especie de ladrillos con los que se construye la búsqueda en los AGs porque éstos contienen las características que hacen a un individuo mejor que otros. Un individuo bueno genera otro también bueno cuando transmite el esquema que lo hace ser relevante dentro de la población.

La probabilidad de transmisión que tienen los esquemas de generación en generación es lo que hace que los algoritmos genéticos sean un método de búsqueda estocástico. Goldberg demuestra que al realizarse cada ciclo del AG la probabilidad de transmisión de los buenos esquemas se incrementa de forma geométrica²⁶. El fundamento matemático es el siguiente²⁷:

Parámetros:

$t =$ Generación sobre la cual se hace el análisis.

$H =$ Identificación del esquema analizado.

²⁶ GOLDBERG E., David. Op. cit., p.30

²⁷ GOLDBERG E., David. Op. cit., p.28

$f(H)$ = Valor de aptitud promedio de los cromosomas que contienen el esquema H .

f_i = Valor de aptitud del cromosoma i .

$m(H, t)$ = Cantidad de cromosomas con esquema H en la generación t .

n = Tamaño de la población

Demostración:

La probabilidad de ser seleccionado para cualquier individuo i , si se usa una ruleta rusa, es igual a:

$$f_i / \sum_{\forall i} f_i \quad (1)$$

como se vio en el numeral 1.5 de este documento. Entonces la probabilidad de que se escoja a cualquier individuo que contenga el esquema H es igual a:

$$f(H) / \sum_{\forall i} f_i \cdot (2)$$

Según lo anterior, la cantidad esperada de individuos con el esquema H dentro de su codificación en la generación $(t + 1)$ será igual a:

$$m(H, t + 1) = m(H, t) \cdot n \cdot f(H) / \sum_{\forall i} f_i \quad (3)$$

Si se tiene que:

$$\sum_{\forall i} f_i / n = \bar{f} \quad (4)$$

donde \bar{f} es el valor promedio de las aptitudes de todos los cromosomas, al reemplazar (4) en (3) se obtiene:

$$m(H, t + 1) = m(H, t) \cdot \frac{f(H)}{\bar{f}} \quad (5)$$

Luego sí el esquema H proporciona una mejor aptitud a los cromosomas que lo contienen, estos tendrán una mayor probabilidad de ser escogidos y habrá más de ellos en la siguiente generación. Por el contrario si el esquema H no provee a los individuos que lo contienen un buen valor de aptitud, la probabilidad de ser seleccionados disminuye y habrá cada vez menos de ellos en las generaciones subsecuentes.

Ahondando aún más en la forma como crecen o decrecen los esquemas se puede demostrar que éstos crecen geométricamente si el valor de aptitud promedio de los individuos que contienen un esquema particular es mayor al valor de aptitud promedio de toda la población. Suponiendo el valor promedio de la aptitud de los cromosomas con esquema H siempre superior al promedio de todas las aptitudes, de la forma $f(H) = \bar{f} + c\bar{f}$, al reemplazar en (5) se obtiene:

$$m(H, t + 1) = m(H, t) \cdot \frac{(\bar{f} + c\bar{f})}{\bar{f}} = m(H, t) \cdot (1 + c) \quad (6)$$

Al partir de una cantidad de individuos conocida en $t = 0$ igual a $m(H, 0)$ se puede reescribir la ecuación para cualquier generación t de la siguiente forma:

$$m(H, t) = m(H, 0) \cdot (1 + c)^t \quad (7)$$

La ecuación (7) tiene la forma de una progresión geométrica lo cual lo indica que la cantidad de individuos con el esquema H crecerá geométricamente dentro de la población si se mantiene que $f(H) = \bar{f} + c\bar{f}$.

La mayor probabilidad de transmitir la información que tienen los cromosomas con esquemas que los hacen fuertes, demuestra como se orienta la búsqueda en un AG. Los esquemas son el fundamento probabilístico del funcionamiento de los algoritmos genéticos.

2. DESCRIPCIÓN DEL PROBLEMA

2.2 IDENTIFICACIÓN DEL PROBLEMA

Existen diversos escenarios posibles bajo los cuales se puede plantear un proyecto inmobiliario. La diversidad de escenarios radica en la capacidad de riesgo del proyectista y la forma de adquisición o negociación del lote. El nivel de riesgo del inversionista depende de su capacidad para endeudarse o para invertir. Las formas de adquisición o negociación del lote dependen de cada caso particular, pero existen ciertas costumbres.

La construcción es una actividad que requiere de gran cantidad de dinero. La capacidad de inversión, o endeudamiento, es definitiva para la viabilidad de un proyecto. En un escenario donde la disponibilidad de dinero este garantizada, la meta del inversionista es generar la mayor utilidad sobre la inversión. En otro escenario donde la disponibilidad de dinero es restringida, el constructor busca obtener utilidad pero conciente de que las ventas deben apalancar financieramente al proyecto. En este caso el objetivo no es maximizar la utilidad sino minimizar la inversión, de forma que el proyecto sea viable.

El otro factor que define el escenario en el que se enmarca un proyecto es la forma de adquisición del lote. Existen varias costumbres mercantiles para la negociación de los lotes entre propietario y promotor inmobiliario. Existen primordialmente dos formas de negociar los lotes. Una es cuando el propietario del lote aporta el lote al proyecto inmobiliario y entra como inversionista. La otra es cuando el terreno hace parte del inventario del proyectista, o éste compra un lote. En el caso de la compra del lote, se pacta una forma de pago y todo el riesgo corre por cuenta del promotor del proyecto. En el caso en que el dueño del lote lo aporta como inversión, el riesgo es compartido.

Durante la crisis económica que se presento en el país en los últimos años de la década del 90, el sector de la construcción se vio seriamente afectado. La liquidez de los constructores

y su capacidad de riesgo se vio minada por el entorno económico. En esta época fue muy popular una figura bajo la cual el propietario de un lote lo aportara como inversión a un proyecto inmobiliario. Debido a la situación económica, muchos propietarios de lotes dejaban desarrollar los proyectos en su propiedad pero no dejaban que este se constituyera como garantía ante una entidad financiera. Bajo ese esquema se hacían inviables muchos proyectos pues un constructor ilíquido y sin capacidad de endeudamiento no tenía la capacidad para promover un proyecto inmobiliario.

Este trabajo de grado es una herramienta para enfrentar esos momentos difíciles. Esos momentos en que el constructor busca alguna posibilidad para desarrollar su trabajo pero las condiciones le son adversas. Momentos en los cuales la maximización de la utilidad no es una premisa, pero si lo es buscar la viabilidad de un proyecto.

Existe otro escenario, más común durante la reactivación económica, en el cual tanto el constructor como el dueño del lote tienen la capacidad para asumir riesgo. En este caso se busca maximizar la utilidad. Este trabajo de grado también es una herramienta para este escenario.

Los proyectos inmobiliarios tienen ciertas características. Los ingresos de dinero se dan por las ventas de unidades residenciales y los egresos se dan debido a los costos directos e indirectos asociados a la construcción de la edificación. El ritmo de ventas y el ritmo de entregas condicionan el ingreso y el egreso de dinero en un proyecto.

El ritmo de ventas proviene del “know how” del proyectista según su experiencia y análisis de información histórica. A partir del ritmo de ventas y la forma de pago de las unidades a vender, se constituye el ingreso. El ritmo de ventas le indica al proyectista cuando va a recibir las cuotas iniciales, mientras los plazos de entrega le indican el ritmo de construcción y el momento en el cual recibe el monto restante. La cuota inicial corresponde a un porcentaje del valor de venta del inmueble y se recibe al momento de cerrar la venta de la unidad. El monto restante por lo general proviene de un crédito que el comprador de la

vivienda asume en un banco y éste a su vez desembolsa al constructor; éste ingreso ocurre cuando se entrega el inmueble al propietario. Por ejemplo si la forma de pago es: 30% de cuota inicial y 70% al momento de la entrega, el ingreso se daría en dos momentos. El 30% del valor de la venta ingresaría en el momento de la venta, y el 70% ingresaría al momento de la entrega del inmueble

El egreso se da según el ritmo de construcción, impuesto por las obligaciones en que incurre el proyectista al comprometerse con los plazos de entrega de los inmuebles. Los egresos corresponden al costo asociado a la construcción de la edificación. Hay que hacer claridad en que el valor de venta es superior al costo de construcción; de no ser así no habría ninguna utilidad sobre el proyecto.

La programación de obra tiene un impacto directo sobre las finanzas del proyecto. Dependiendo el escenario bajo el cual se enmarca el proyecto, el objetivo financiero es diferente. Adicionalmente las obligaciones con los clientes tienen impacto sobre la programación de la obra. Este trabajo de grado conjuga todos esos factores, para dos escenarios bien definidos, con una herramienta que le proporciona al constructor una programación de obra acorde con el objetivo financiero y las obligaciones con los clientes.

2.2 PLANTEAMIENTO DEL MODELO MATEMÁTICO

Están definidos dos escenarios diferentes, con objetivos diferentes. Un objetivo es minimizar la inversión, el otro es maximizar la utilidad. Así pues se define el problema para lograr dichos objetivos:

Conjuntos:

$A : \{1,2,3,\dots,n\}$ Conjunto de actividades que componen el proyecto.

$P : \{1,2,3,\dots,m\}$ Conjunto de períodos en los que se construye el proyecto.

$Z_a \quad Z_a \subseteq A \quad \forall a$ Conjunto de actividades que preceden la actividad a.

$K\{1,2,3,\dots,l\}$ Conjunto de tipos de mano de obra disponibles.

$J\{1,2,3,\dots,y\}$ Conjunto de recursos materiales disponibles.

Parámetros:

$I_p =$ Ingreso en el período p

$D_{j,a} =$ Demanda del recurso material j para la actividad a

$U_j =$ Valor unitario del recurso material j

$R_{k,a} =$ Rendimiento de la mano de obra k en la actividad a

$V_k =$ Valor unitario de la mano de obra k

$Q_a =$ Cantidad a ejecutar en la actividad a

$M_a =$ Duración máxima de la actividad a

$N_a =$ Duración mínima de la actividad a

$i =$ Tasa de oportunidad del inversionista

Variables:

$m_{a,k} =$ Demanda de mano de obra k para la actividad a

$d_a =$ Duración de la actividad a

$c_{a,p} =$ Costo asociado a la actividad a para el periodo p

$s_a =$ Fecha inicial de la actividad a

$f_a =$ Fecha final de la actividad a

$e_p =$ Egreso en el período p

$fc_p =$ Flujo de caja neto al final del período p

$fca_p =$ Flujo de caja acumulado al final del período p

$cd_a =$ Costo directo de la actividad a

Modelo:

$$\text{Maximizar } \text{Min}(fca_p) \quad (\text{A})$$

$$\text{Maximizar } \sum_{p=1}^m \frac{fc_p}{(1+i)^p} \quad (\text{B})$$

Sujeto A:

$$fca_p = fca_{p-1} + I_p + e_p \quad (2a)$$

$$fc_p = I_p + e_p \quad (2b)$$

$$e_p = -\sum_{\forall a} c_{a,p} \quad (3)$$

$$c_{a,p} = 0; \forall p < s_a \quad (4a)$$

$$c_{a,p} = \frac{cd_a}{(f_a - s_a)} \times (p - s_a); \forall p / f_a \geq p \geq s_a \quad (4b)$$

$$c_{a,p} = 0; \forall p > s_a \quad (4c)$$

$$cd_a = \sum_{\forall j} (D_{j,a} \times U_j) + \sum_{\forall k} (m_{a,k} \times V_k) \quad (5)$$

$$M_a \geq d_a \geq N_a \quad (6)$$

$$m_{a,k} = \frac{Q_a}{(f_a - s_a)} \times \frac{1}{R_{k,a}} \quad (7)$$

$$f_a - s_a > 0 \quad (8)$$

$$s_a - f_b \geq 0; \forall b \in Z_a \quad (9)$$

$$fca_0 = I_0$$

Donde:

Se supone que el costo asociado a cada actividad se causa proporcionalmente durante el tiempo de desarrollo de la misma. Se supone que el costo unitario de la mano de obra y de los recursos materiales es constante para todos los períodos. Se supone mano de obra y recursos materiales ilimitados.

La función objetivo (A) es para atender el escenario en el cual la capacidad financiera del constructor es mínima. Se maximiza el valor del mínimo flujo de caja acumulado para garantizar que se requiera la menor o ninguna inversión en el proyecto.

La función objetivo (B) atiende el caso en el cual se busca maximizar la utilidad. Para esta situación se maximiza el Valor Presente Neto del flujo de caja neto en cada período.

La eq. (2a) indica que es el flujo de caja acumulado para un período, el cual está compuesto por el flujo de caja neto del periodo anterior, los ingresos en el período P y los egresos en el

mismo período. La ecuación (2b) muestra como se obtiene el flujo de caja neto de cada período.

El propósito de la familia de ecuaciones (4) es calcular el costo directo causado por cada actividad según el momento en que se haga la evaluación. Por ejemplo si el momento en que se evalúa la función es un momento previo a la iniciación de la actividad, entonces no habrá ningún costo causado por dicha actividad. Lo mismo sucede si el momento en que se evalúa la función es posterior a la finalización de la actividad. Existe otra situación en la cual la actividad se este llevando a cabo en el momento en que se quiere evaluar la función. Para este caso se hace una suposición fuerte, según la cual el egreso de la actividad se causa proporcional a la duración de la misma. Esta suposición es valida dado que la evaluación de la función descrita en la eq. (4b), se hará mes a mes. Sucede que al hacer la evaluación mes a mes, lo que equivale a dos períodos de quince días o a cuatro semanas, se están contemplando periodos de pago ya sea mensuales quincenales o semanales²⁸.

La eq. (5) indica la forma como está compuesto el costo directo de cada actividad. Dicho costo depende de la demanda de recurso y el valor unitario del mismo. También es función de la cantidad de mano de obra necesaria para llevar a cabo la actividad en el tiempo previsto (eq. 7) y del valor unitario de tal mano de obra.

Las restricciones limitan las soluciones a la factibilidad del resultado que representan. La restricción en la eq. (9) indica que entre actividades con relación de precedencia no puede

²⁸ Por ejemplo si una actividad inicia el 10 de Abril de 2005, la evaluación de la función se hará el día 30 del mismo mes. Se supone que el costo por mano de obra es de 30 unidades pagaderas en 8 semanas y que los materiales tienen un costo de 70 unidades pagaderas en 3 quincenas (30 de Abril, 15 de Mayo y 31 de Mayo). Dicha actividad tiene un costo directo total de 100 unidades monetarias y una duración de 50 días, entonces al 30 de Abril habrán corrido 20 días de trabajo y se habrán causado egresos por 40 ($20 \cdot 100 / 50$) unidades monetarias. De las 40 unidades monetarias se habrán pagado 12 en mano de obra y 28 en materiales según la suposición realizada. En términos de obligaciones reales con proveedores y trabajadores a ese día habrán corrido 3 semanas lo cual equivaldría a un egreso de 11.25 ($30 / 8 \cdot 3$) unidades monetarias por mano de obra y a 23.33 ($70 / 3 \cdot 1$) unidades monetarias por compra de materiales. Ambos valores son muy parecidos con los de la suposición, pero es que además el día 31 de mayo ya se habrá causado el costo directo total de la actividad, es decir 100 unidades monetarias, lo cual es perfectamente real.

darse el inicio de una actividad sin que haya finalizado su predecesora. Pero si puede darse el caso de que una actividad no inicie inmediatamente termine su predecesora.

Es necesario que las actividades además de cumplir con sus relaciones de precedencia se desarrollen en un tiempo determinado. Para esto en la eq. (6), se ha restringido dicha duración ente un máximo y un mínimo de duración previamente establecidos para cada actividad.

2.3 CONSIDERACIONES PREVIAS DEL MODELO MATEMÁTICO

Se investigaron dos formas de expresar la función objetivo antes de llegar a la que se empleó finalmente para el escenario donde el inversionista busca maximizar el valor del mínimo flujo de caja acumulado.

La primera aproximación que se hizo fue basada en una regresión lineal. Se quería ajustar la serie de los egresos con la serie de los ingresos mes a mes. El resultado de un ejercicio de ese estilo produce una nivelación entre el ingreso y el egreso que no es deseable pues no solo se busca que el ingreso y el egreso sean parecidos, también se espera que hayan soluciones en donde el ingreso supere al ingreso en todos los períodos.

Otra consideración que se hizo fue la maximización de la suma de los flujos de caja acumulados de cada periodo. El problema con esta consideración es la posibilidad de que el flujo de caja de un periodo donde los egresos superen ampliamente los ingresos se anule al sumarse con el flujo de caja de un período donde los ingresos superan considerablemente los egresos. Si se presenta esa situación, la sumatoria de los flujos de caja netos no proporciona suficiente información para saber que tan grande era la necesidad de dinero en el peor período.

2.4 PARTICULARIDADES DEL PROBLEMA PARA EL CASO COLOMBIANO

Los dos escenarios para los cuales se ha desarrollado este trabajo de grado, están determinados por las condiciones sociales y económicas. El escenario donde el constructor no tiene capacidad de crédito ni de inversión esta determinado por condiciones económicas muy claras y propias de países en recesión. Mientras que el escenario en el cual se busca maximizar la utilidad sobre las ventas se da en un entorno de reactivación económica en donde exista capacidad de riesgo por parte tanto del promotor del proyecto como del propietario del lote.

Los cambios sucesores a la apertura económica de 1991 cambiaron la realidad del país, y también la mentalidad de su gente. En el año 2002 se realizo un ejercicio de planeación por escenarios para el sector de la construcción en el seno de la Cámara Colombiana de la Construcción (CAMACOL)²⁹. En ese trabajo se identificaron los determinantes de la situación del sector y se plantearon las perspectivas de cara a la nueva realidad.

Los determinantes de la situación del sector de la construcción tienen raíz en los antecedentes históricos del país. La dinámica inflacionaria de la economía colombiana, común desde los años setenta, afecto los precios de la finca raíz; tanto así que el precio de algunos bienes era superior en Colombia a el precio de un bien con las mismas características en Estados Unidos ³⁰. Existía un paradigma, se pensaba que el valor de la finca raíz se mantendría a pesar de cualquier vicisitud. Muchas constructoras incrementaron su inventario de tierras, y muchas personas compraron inmuebles pues pensaban que su inversión estaba garantizada. La insostenibilidad de la situación afloro en 1995, su reflejo fueron los precios inasequibles de las viviendas, y la congelación de los inventarios de tierras. Otros ingredientes de la realidad nacional, también surgieron su efecto. El narcotráfico, la guerrilla, la crisis fiscal y de gobernabilidad, incrementaron los niveles de desconfianza y deterioraron la economía. En 1998 el Banco de la Republica tuvo que

²⁹ MANRIQUE RUIZ, Francisco. Señales del Futuro, Escenarios para La Industria de la Construcción. Presidencia CAMACOL. Bogotá: CAMACOL, 2002.

³⁰ Ibid., p. 7.

intervenir para proteger la banda cambiaria. Producto de ello las tasas de interés se incrementaron, lo que afectó de manera significativa al sector productivo y a las empresas con altos niveles de endeudamiento. Las constructoras que habían utilizado su capacidad de endeudamiento para adquirir terrenos sufrieron con creces los efectos de la crisis. La desvalorización de sus inventarios sumado al incremento en las tasas de interés produjo un efecto devastador. El valor de los activos, constituido en su mayoría por terrenos, disminuyó mientras que el valor del pasivo se incrementó.

Muchas empresas quebraron y muchas otras se adaptaron a la situación. Las que tuvieron la capacidad y la decisión de afrontar la nueva realidad se vieron obligadas a promover y construir proyectos con el menor nivel de endeudamiento o inversión posible. Para ese momento la compra de lotes se constituía como una práctica obsoleta, entonces se hizo popular la figura bajo la cual el propietario de un lote lo aportaba para desarrollar en él un proyecto cuya promoción y construcción estaba a cargo de una empresa constructora.

El año 2002 marcó el inicio de la reactivación económica con la elección de un nuevo gobierno. La figura del aporte de terreno en los proyectos inmobiliarios se tornó costumbre. Muchas empresas constructoras retomaron su capacidad de inversión y de endeudamiento sin embargo la crisis dejó como enseñanza que la compra de terrenos no es una práctica saludable financieramente. Es más rentable utilizar la inversión o el endeudamiento para la construcción de proyectos y no para la adquisición de un inventario de tierras. La nueva realidad, y las nuevas posibilidades enmarcan un escenario bajo el cual se busca maximizar la utilidad sobre la inversión o el endeudamiento, aplicando lo aprendido durante la crisis.

3. DESARROLLO DEL ALGORITMO GENÉTICO

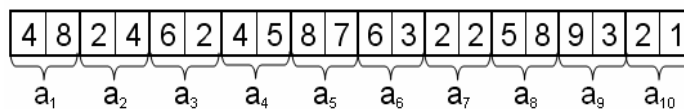
3.1 CODIFICACIÓN DEL CROMOSOMA

Se ha optado por codificar en el cromosoma dos variables por actividad, en parámetros reales. Dichas variables son la duración de la actividad a [da] y el tiempo que transcurre desde que finaliza la actividad predecesora hasta que inicia la actividad a [Ga]. Se seleccionó este tipo de codificación ya que todas las soluciones que representa son factibles, lo cual implica que cualquier valor que tomen estas variables para cualquier actividad representa una solución factible con respecto a la totalidad del problema.

El cromosoma puede construirse mediante la implementación de tres codificaciones diferentes a partir de las dos variables que se van a incluir. Aunque cada tipo de codificación tiene implicaciones diferentes en el funcionamiento del algoritmo.

3.1.1 Codificación plana con alelos consecutivos. Este tipo de codificación implica situar ambas variables consecutivamente para cada actividad. Suponiendo un proyecto con 10 actividades se tendría el siguiente cromosoma:

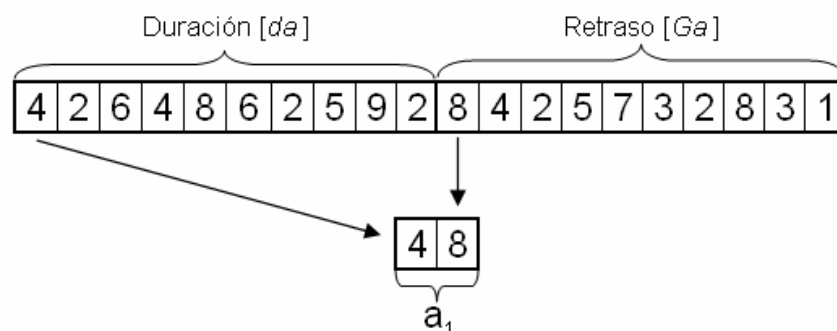
Figura 10. Codificación con alelos consecutivos



Para la actividad a_1 se tendría $da = 4$ y $Ga = 8$, es decir que a_1 tiene una duración de 4 días y un retraso de 8 días. Con esta codificación se tiene una longitud corta del esquema ya que las variables están una enseguida de la otra.

3.1.2 Codificación plana con alelos no consecutivos. La codificación con alelos no consecutivos se refiere a listar primero una variable para todas las actividades y luego listar la otra variable. Con esta implementación el mismo cromosoma del numeral 3.1.1 quedaría así:

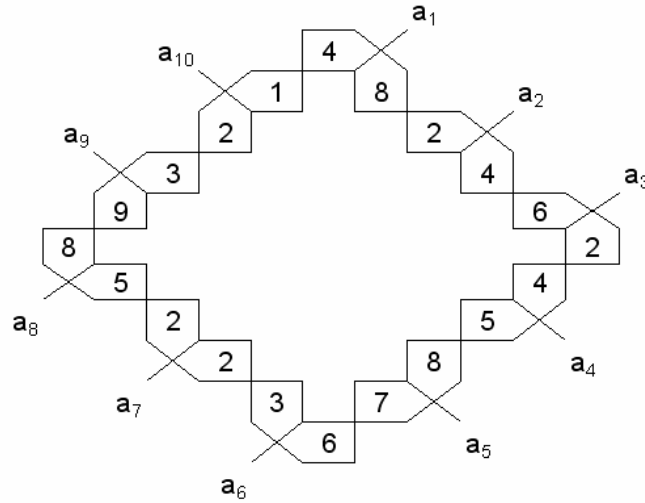
Figura 11. Codificación con alelos no consecutivos



Esta codificación implica una longitud del esquema mayor a la anterior, ya que si se quiere conservar la característica de alguna actividad es necesario conservar toda la información codificada entre los valores que le corresponden a la actividad. Por ejemplo si se quiere conservar la característica de a_4 (4,5), el esquema sería: (4,8,6,2,5,9,2,8,4,2,5).

3.1.3 Codificación circular. La codificación circular puede hacerse con alelos consecutivos o no. La diferencia con ésta es que incluye una posición más que se puede aprovechar en la recombinación. Con cualquiera de las codificaciones anteriores se tienen $(n - 1)$ puntos de cruce para hacer la recombinación, mientras con la implementación circular se tienen (n) puntos de cruce. Esto se debe a que el cromosoma es de la siguiente forma:

Figura 12. Codificación circular con alelos consecutivos



Al ser circular el cromosoma, existe una posición adicional que es la que conecta el último alelo con el primero. Otra implicación de este cromosoma es que sólo se le puede operar con recombinación de dos o más puntos ya que una recombinación de un solo punto de cruce simplemente rompería el círculo pero no partiría el cromosoma en partes.

La longitud del esquema dependerá, al igual que en las otras implementaciones, solamente del orden en que se sitúen los alelos. Pueden situarse los alelos de forma consecutiva o no; de modo consecutivo la longitud del esquema es corta, sino la longitud del esquema es más larga.

3.1.4 Impacto de la codificación en el operador de recombinación. El tipo de codificación que se emplee tiene su impacto en la recombinación debido a su ingerencia en la probabilidad de transmisión del esquema. Dicha probabilidad depende de la longitud del esquema, de la cantidad de puntos de cruce utilizados en la recombinación y de la cantidad de posiciones disponibles para ubicar los puntos de cruce.

Para explicar bien la ingerencia de los tipos de codificación vistos, es necesario plantear matemáticamente la probabilidad que tiene un esquema de ser transmitido luego de la

recombinación dependiendo del tipo de codificación y de la cantidad de puntos de cruce. El tipo de codificación influye sobre la cantidad de posiciones con que se cuenta y la longitud del esquema.

Parámetros:

n = Tamaño del cromosoma (cantidad de genes)

L = Longitud del esquema

Codificación plana y recombinación con un solo punto de cruce:

Cuando se tienen alelos consecutivos y representación plana o no circular existen $(n-1)$ posiciones susceptibles de ser puntos de cruce. Si se quiere conservar un esquema de longitud L , entonces habrá solo $((n-1)-(L-1))$ posiciones disponibles para situar los puntos de cruce.

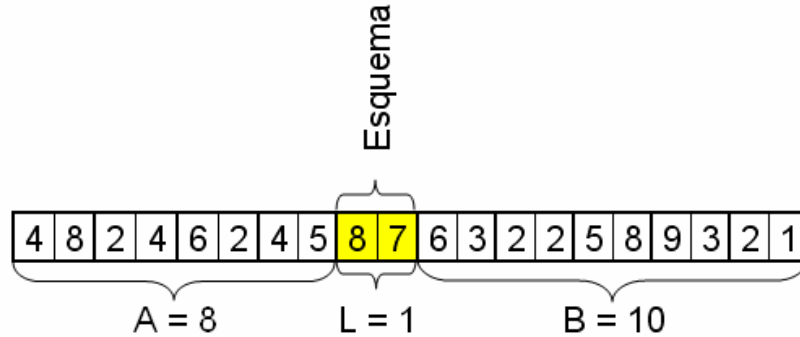
Cuando se emplea recombinación con un solo punto de cruce sobre un cromosoma plano, la probabilidad de escoger un punto por fuera del esquema para que éste se pueda transmitir está dada por la ecuación:

$$\frac{(n-1)-(L-1)}{(n-1)} = 1 - \frac{(L-1)}{(n-1)} \quad (1)$$

Codificación plana y recombinación con dos puntos de cruce:

Cuando la recombinación es de dos puntos, la probabilidad está condicionada a que ambos puntos se escojan por fuera del esquema. Es mejor observar un cromosoma ejemplo para ubicar bien los puntos de cruce:

Figura 13. Puntos de cruce disponibles en un cromosoma



En la región A habrá 8 puntos de cruce posible, en la región L habrá $(L-1) = (2-1) = 1$ punto y en la región B habrá 10. La probabilidad de que se transmita el esquema será:

$$\left(\frac{(20-1)-1-8}{(20-1)} + \frac{(20-1)-1-10}{(20-1)} \right)^2 = 0.22$$

De forma general sería:

$$\left(\frac{(n-1)-(L-1)-B}{(n-1)} + \frac{(n-1)-(L-1)-A}{(n-1)} \right)^2 = \left(\frac{(2n-2L-B-A)}{(n-1)} \right)^2 \quad (2)$$

La ecuación (2) permite calcular la probabilidad para que se escojan dos puntos por fuera del esquema pues al multiplicar por si misma la probabilidad para que un solo punto escogido caiga en A o en B, se obtiene la probabilidad condicional para que al escoger dos puntos ambos caigan en A o B. Otra forma de escribir la ecuación (2) es:

$$= \left(1 - \frac{(L-1)}{(n-1)} \right)^2 \quad (3)$$

Esta ecuación permite calcular la misma probabilidad de la ecuación (2) pues es el cuadrado de la probabilidad de escoger un punto por fuera del esquema; o lo mismo que la probabilidad condicional de escoger dos puntos por fuera del esquema.

Codificación circular y recombinación con dos puntos de cruce:

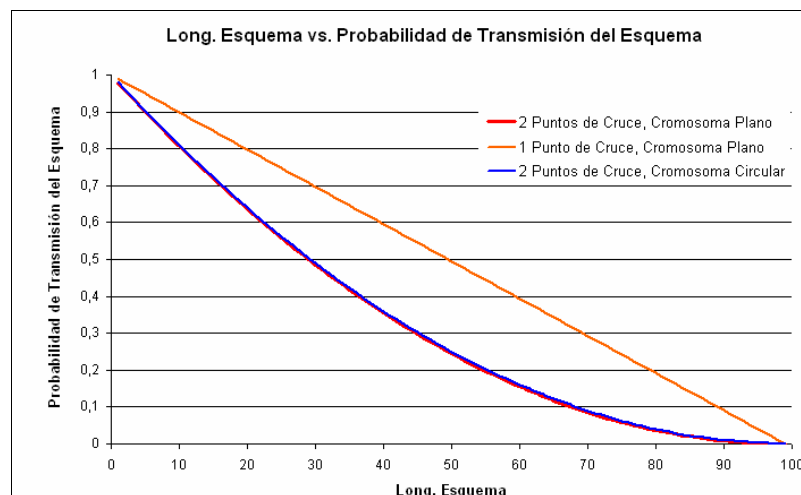
Cuando se trata de un cromosoma circular la probabilidad es diferente porque la cantidad de puntos de cruce posibles es igual a la longitud del cromosoma aunque la cantidad de puntos disponibles, si se quiere transmitir el esquema, es igual a $n - (L - 1)$. Entonces la probabilidad de transmisión del esquema cuando se hace recombinación de dos puntos sobre un cromosoma circular es:

$$\left(1 - \frac{(L-1)}{n}\right)^2 \quad (4)$$

Comparación:

De las ecuaciones (1), (3) y (4) se puede deducir que la probabilidad de transmisión del esquema -para cromosomas de igual longitud- depende sólo de la cantidad de puntos de cruce en la recombinación y de la longitud del esquema. La siguiente gráfica permite confrontar claramente, para un cromosoma de longitud definida, cómo se comporta la probabilidad de transmisión del esquema según su longitud y el tipo de recombinación:

Gráfica 2. Longitud del Esquema vs. Probabilidad de Transmisión del Esquema, para un cromosoma de 100 genes.



En la Gráfica 2 se observa con claridad que entre más largo el esquema, menor es la probabilidad de que éste se transmita a sus sucesores. También es claro que la diferencia entre la codificación circular y la plana no es muy significativa para la probabilidad de transmisión del esquema, cuando se emplea la recombinación con dos puntos de cruce. Lo anterior permite concluir que de las codificaciones vistas la más beneficiosa sería la implementación plana con alelos consecutivos, ya que la longitud del esquema es menor (como se explicó en el numeral 3.1.1 de este documento); por esa razón la probabilidad de que éste perdure a través de las generaciones es mayor.

3.2 TÉCNICA DE SELECCIÓN

En la herramienta que se desarrollo en este trabajo de grado, se emplearon cuatro técnicas de selección:

- Selección aleatoria
- Selección por torneo
- Selección proporcional
- Selección escalada

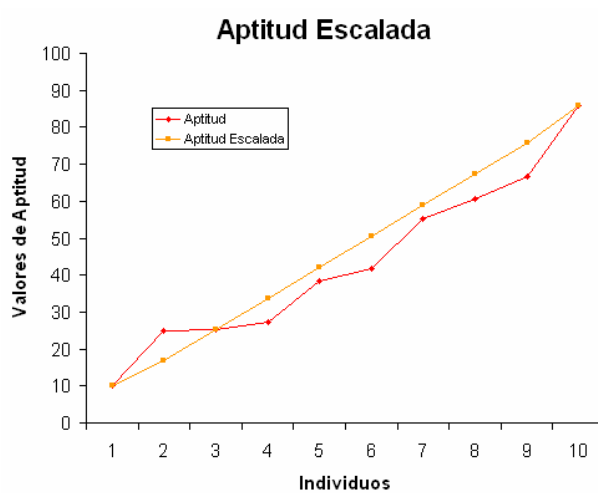
La selección por torneo y la selección proporcional fueron ampliamente explicadas en la sección 1.5, por ser dos operadores de implementación usual en los algoritmos genéticos. La selección aleatoria no tiene ciencia alguna, con esta técnica se escogen padres aleatoriamente sin ningún criterio para recombinarlos y generar nuevos hijos. La selección escalonada por su parte, se basa en la selección proporcional pero tiene unas consideraciones adicionales.

La técnica de selección escalada, propuesta por Goldberg³¹, busca escalar el valor de aptitud de todos los cromosomas padres y luego hacer selección proporcional sobre la población. La forma de hacerlo es trazando una recta entre el menor valor de aptitud y el

³¹ GOLDBERG E., David. Op. cit., p. 76

mayor valor de aptitud en la población, y luego prolongando o truncando los valores de aptitud de cada individuo al valor de la recta que le corresponde. La siguiente Gráfica muestra claramente cómo se hace el escalamiento del valor de aptitud.

Gráfica 3. Escalamiento del valor de aptitud.



El valor de aptitud del individuo 2 se vio truncado, mientras que el valor de aptitud del individuo 6, por ejemplo, se vio prolongado hasta la recta.

La selección escalada sirve para prevenir la convergencia prematura en el algoritmo. Con esta técnica se evita que un individuo fuerte tenga una probabilidad muy alta de ser escogido y que un individuo muy débil tenga una probabilidad muy baja de ser escogido. De esa forma es más fácil mantener la diversidad en la población y se evita la convergencia prematura a un valor mediocre.

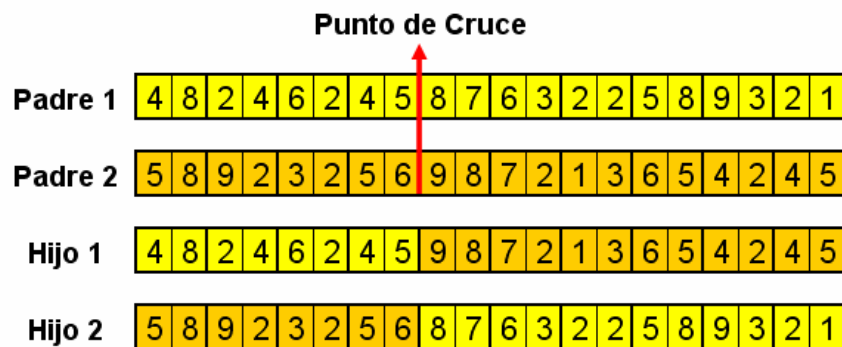
3.3 RECOMBINACIÓN

El operador de recombinación es muy importante en el funcionamiento del algoritmo genético ya que permite la generación de nuevos individuos y además tiene gran influencia

sobre la probabilidad de transmisión del esquema, como ya se vio. Este operador varía dependiendo de la cantidad de puntos de cruce y de la cantidad de padres que aporten información. La implementación de la recombinación se hizo considerando que se pueden generar tantos hijos por recombinación como padres haya involucrados. Entonces para la recombinación con dos padres y un solo punto o dos puntos de cruce máximo se podrán obtener dos hijos por recombinación; para la recombinación con tres padres y dos puntos de cruce máximo se podrán obtener 3 hijos por recombinación.

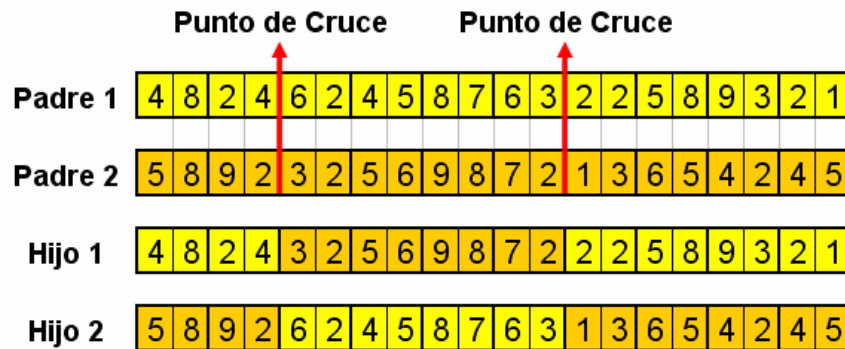
3.3.1 Recombinación con un solo punto de cruce y dos padres. Este tipo de operador funciona partiendo dos cromosomas padres en dos sub-cadenas cada uno y generando dos hijos a partir de ello. Gráficamente es así:

Figura 14. Recombinación con un solo punto de cruce y dos padres



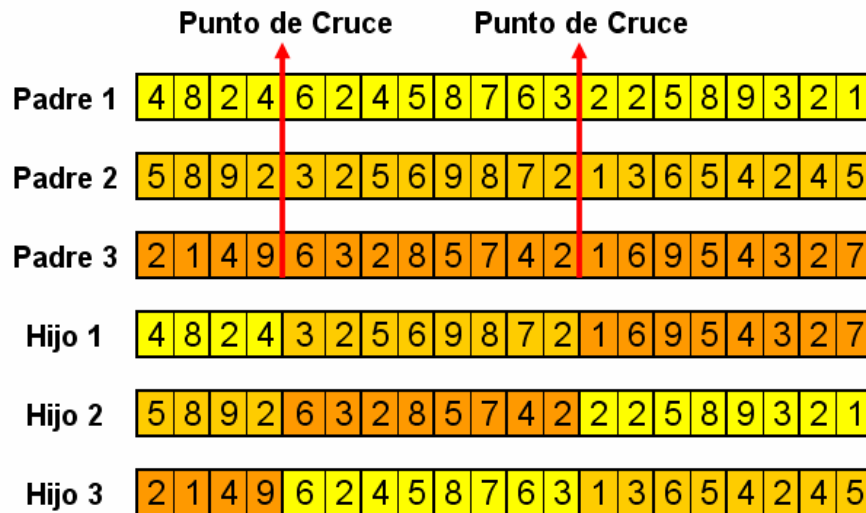
3.3.2 Recombinación con dos puntos de cruce y dos padres. Cuando se tienen dos padres que aportan información y dos puntos de recombinación se generan dos hijos de la siguiente forma:

Figura 15. Recombinación con dos puntos de cruce y dos padres



3.3.3 Recombinación con dos puntos de cruce y tres padres. Con este operador la información de los tres padres se transmite de la siguiente manera:

Figura 16. Recombinación con dos puntos de cruce y tres padres



3.4 MUTACIÓN

La mutación consiste en cambiar de alguna forma el contenido de los cromosomas. Esto se hace con el objetivo de alterar un poco el ambiente de los cromosomas, lo cual a su vez permite explorar nuevas zonas del espacio solución.

3.4.1 Reemplazo aleatorio. En una población de diez individuos, para los cuales cada cromosoma está compuesto por diez alelos, se tendrá un total de cien alelos. Para un porcentaje de mutación igual a dos por ciento, se escogerán aleatoriamente 2 alelos dentro de los cien existentes y se cambiará su valor aleatoriamente.

3.4.2 Incremento – decremento. Este operador funciona de igual forma que el reemplazo aleatorio pero en vez de cambiar el valor aleatoriamente, se incrementa o decrementa el valor de los alelos seleccionados en una unidad.

3.4.3 Mutación Variable. Este operador hace lo mismo que el reemplazo aleatorio pero el porcentaje de mutación se incrementa generación a generación desde un porcentaje mínimo hasta un porcentaje máximo previamente establecidos.

3.5 SELECCIÓN DE LOS INDIVIDUOS QUE COMPONEN UNA NUEVA GENERACIÓN

Una vez un grupo de padres ha generado un grupo de hijos, es necesario escoger entre los mejores padres y los mejores hijos los individuos que serán padres de una nueva generación. Se implementaron dos formas de generar la nueva generación.

3.5.1 Mejores padres mejores hijos. Se organizan en orden descendente todos los valores de aptitud de los padres y los hijos. Se escoge un número de individuos igual al tamaño de la población y los escogidos conforman el grupo de padres de una nueva generación.

3.5.2 Elitismo. Se organizan independientemente los valores de aptitud de los padres y de los hijos en orden descendente. Se escoge un número de padres definido según el porcentaje de elitismo, y un número de hijos tal que completen el tamaño de la población. Los individuos seleccionados serán los padres de una nueva generación.

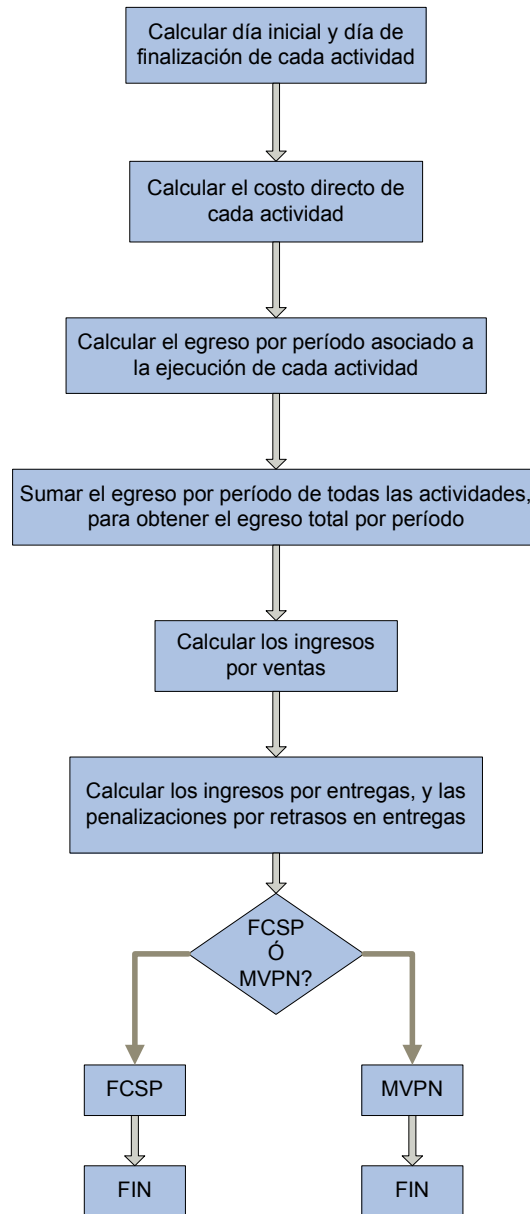
3.6 FUNCIÓN DE APTITUD

Como se dijo en el capítulo 2, el objetivo financiero es diferente según el escenario bajo el cual se enmarca un proyecto. Por eso en la sección 2.2 se plantearon dos funciones objetivo. Una para atender las necesidades de un escenario donde no existe capacidad de inversión o endeudamiento y otra para una situación donde se busca maximizar la utilidad.

El primer escenario mencionado requiere encontrar la programación de obra que proporcione el mayor valor del flujo de caja acumulado mínimo. La función objetivo para este caso se denominó Flujos de Caja Siempre Positivos (FCSP). Para el otro escenario se requiere encontrar la programación de obra que proporcione el mayor valor presente neto de los flujos de caja netos. Esta función objetivo se denominó Maximizar el Valor Presente Neto de los Flujo de Caja Netos (MVPN).

El cálculo del valor de aptitud requiere de varios pasos que están contenidos en el diagrama 1.

Diagrama 1. Diagrama de flujo general de la función de evaluación



Cada uno de los pasos que componen el cálculo del valor de aptitud se desarrollan a continuación.

Diagrama 2. Diagrama de flujo de la función que calcula los días en que inicia y finaliza cada actividad

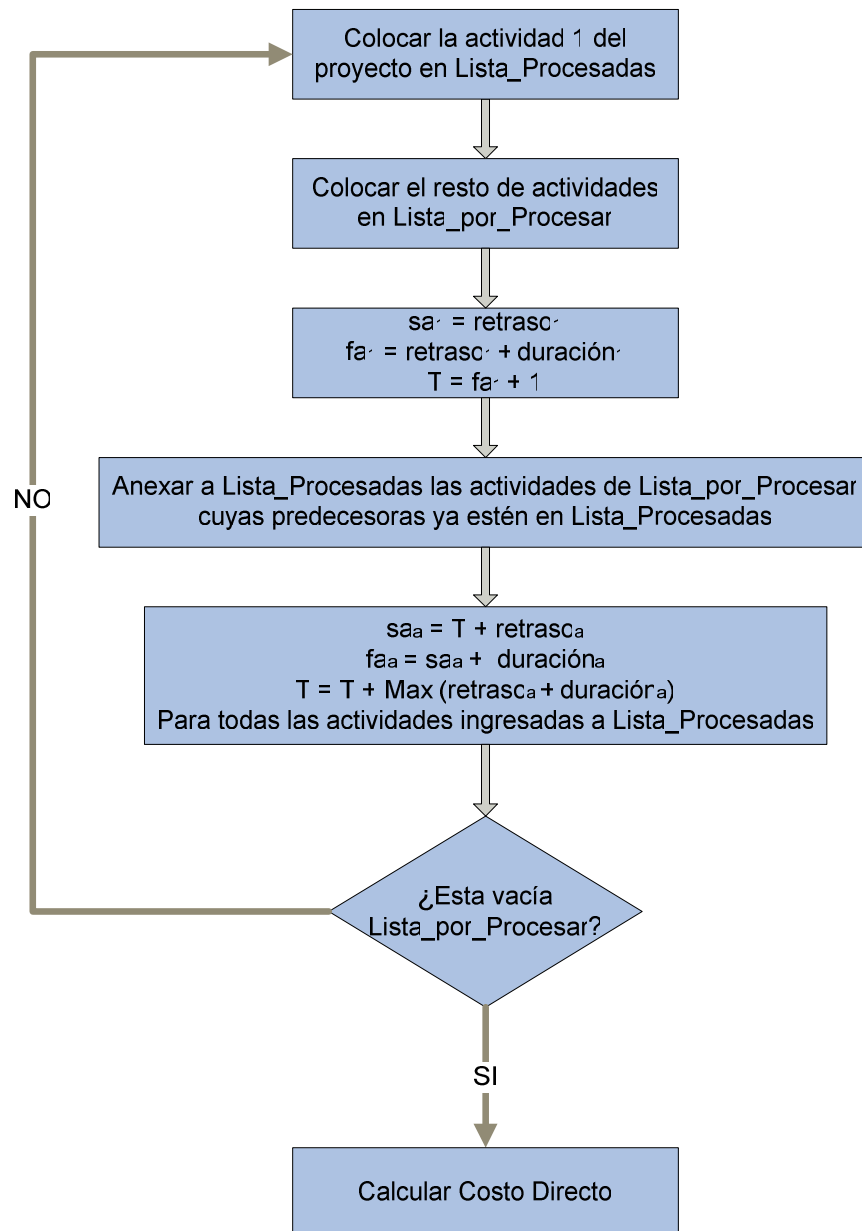


Diagrama 3. Diagrama de flujo de la función que calcula el costo directo de cada actividad

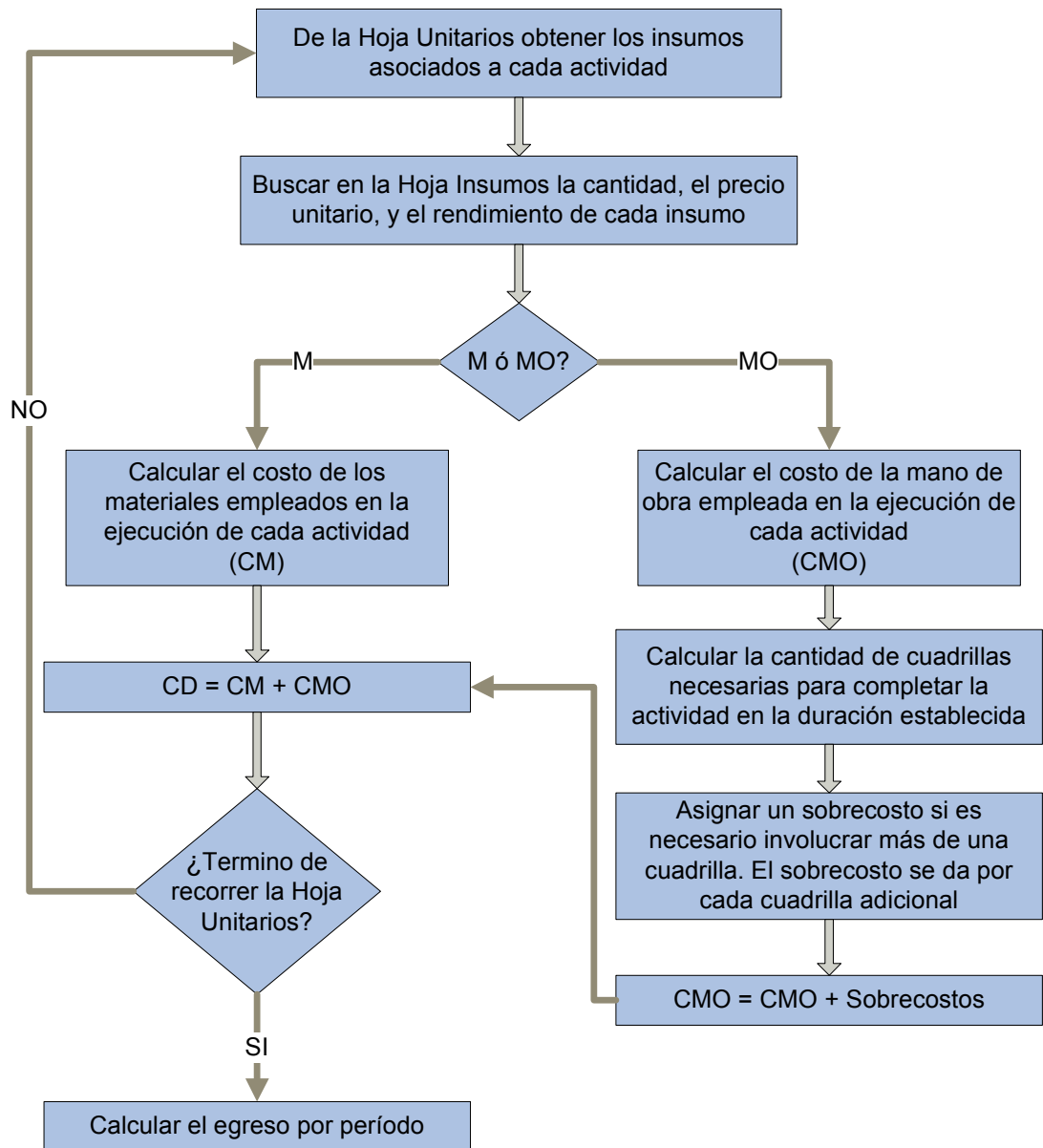


Diagrama 4. Diagrama de flujo la función que calcula los egresos de dinero por período asociado a cada actividad y el egreso de dinero total por período

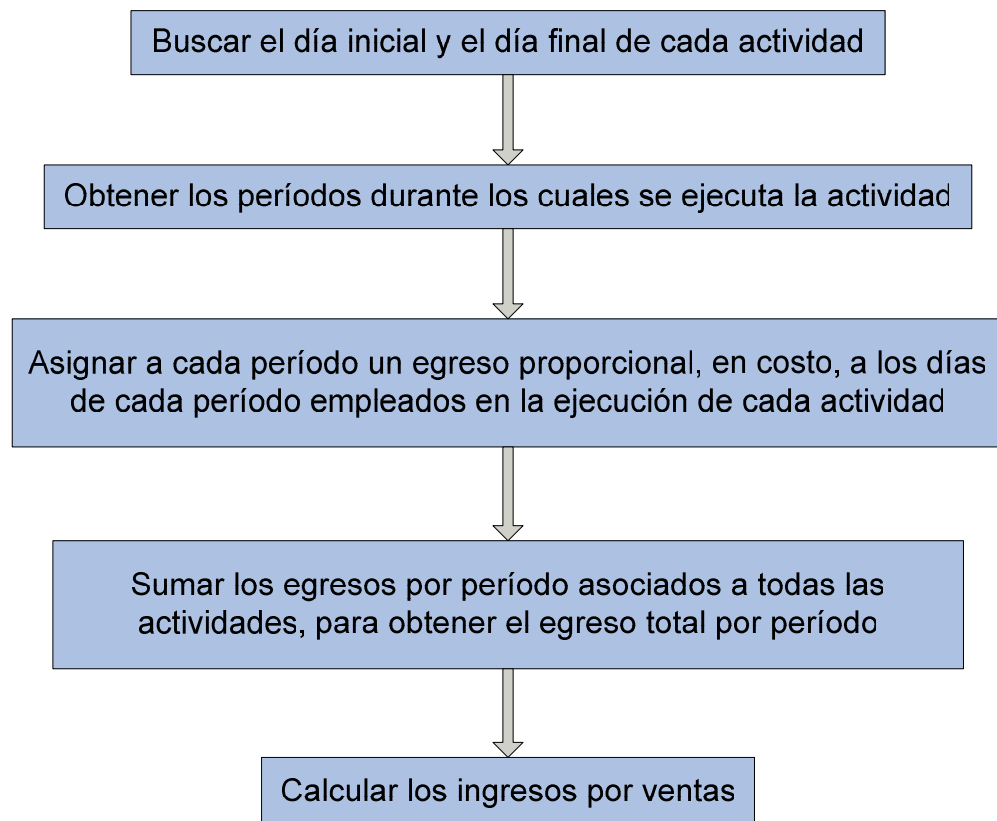


Diagrama 5. Diagrama de flujo de la función que calcula los ingresos de dinero asociados a las ventas

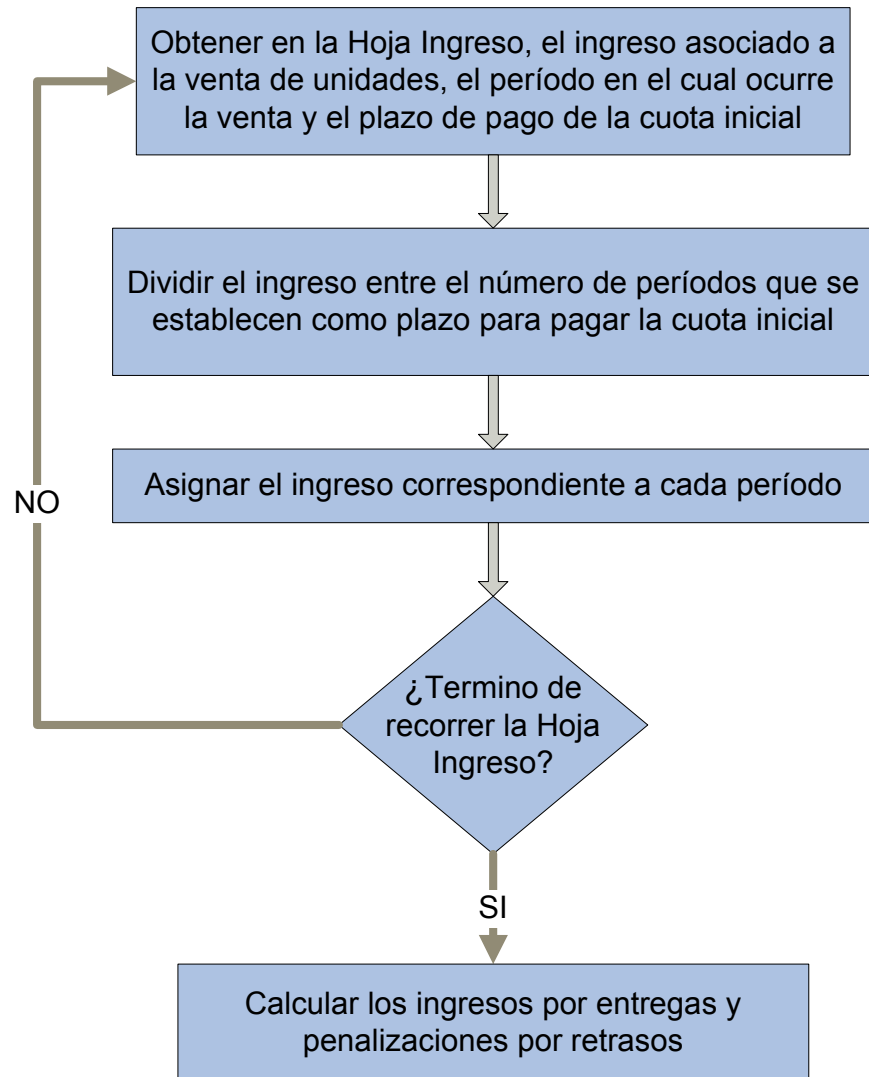


Diagrama 6. Diagrama de flujo de la función que calcula los ingresos de dinero asociados a las entregas y las penalizaciones por retrasos

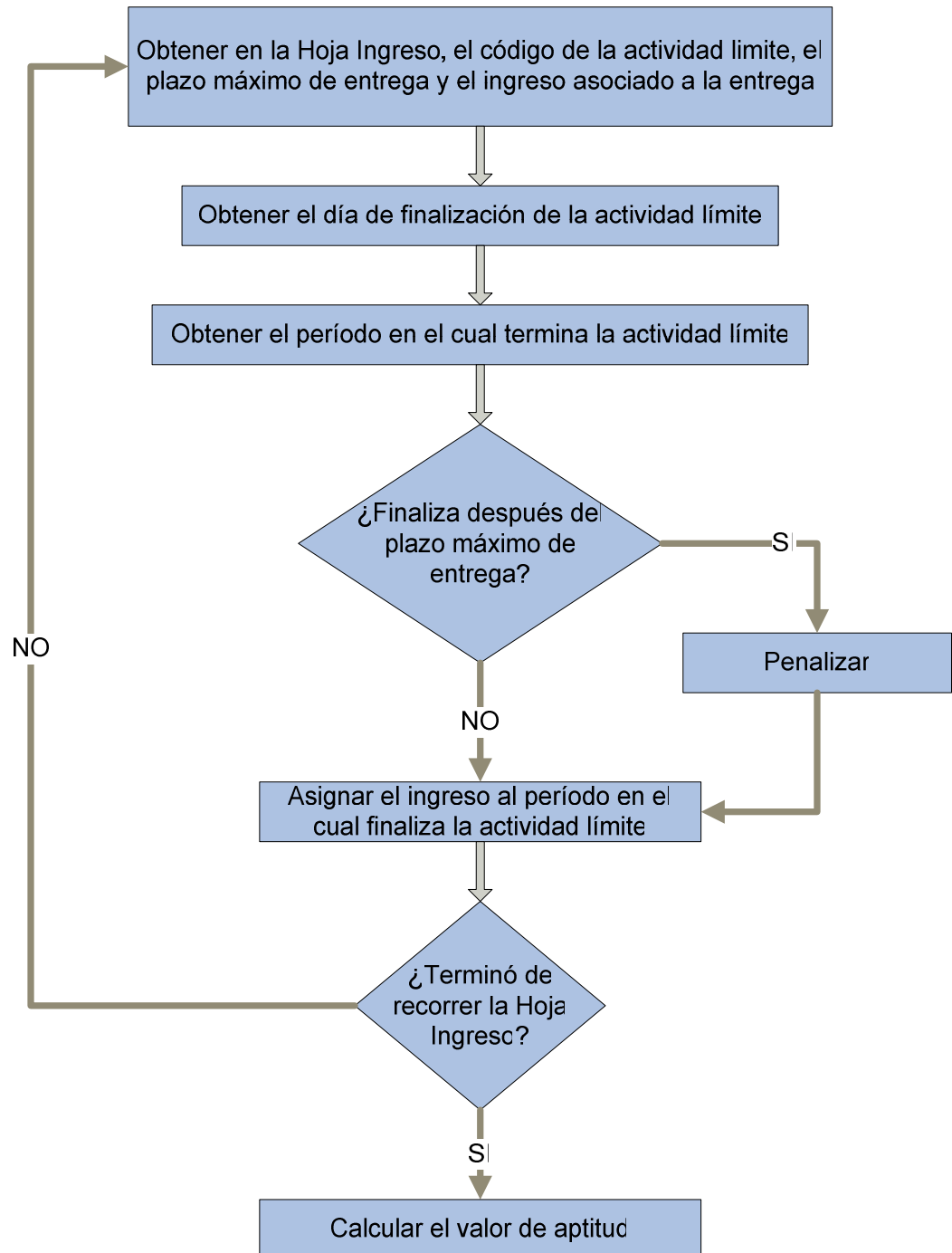


Diagrama 7. Diagrama de flujo de la función que calcula el valor de aptitud para MVPN

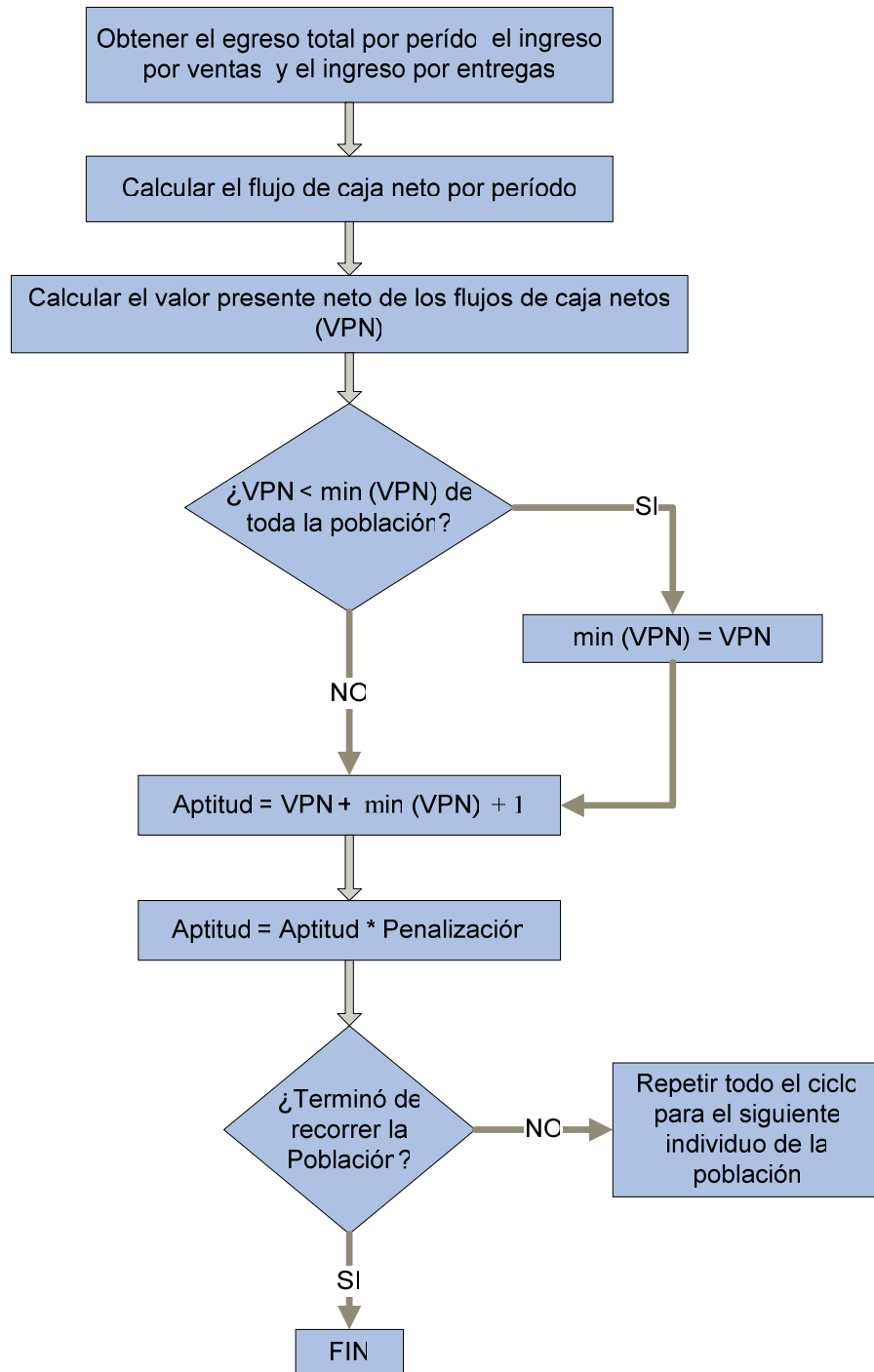
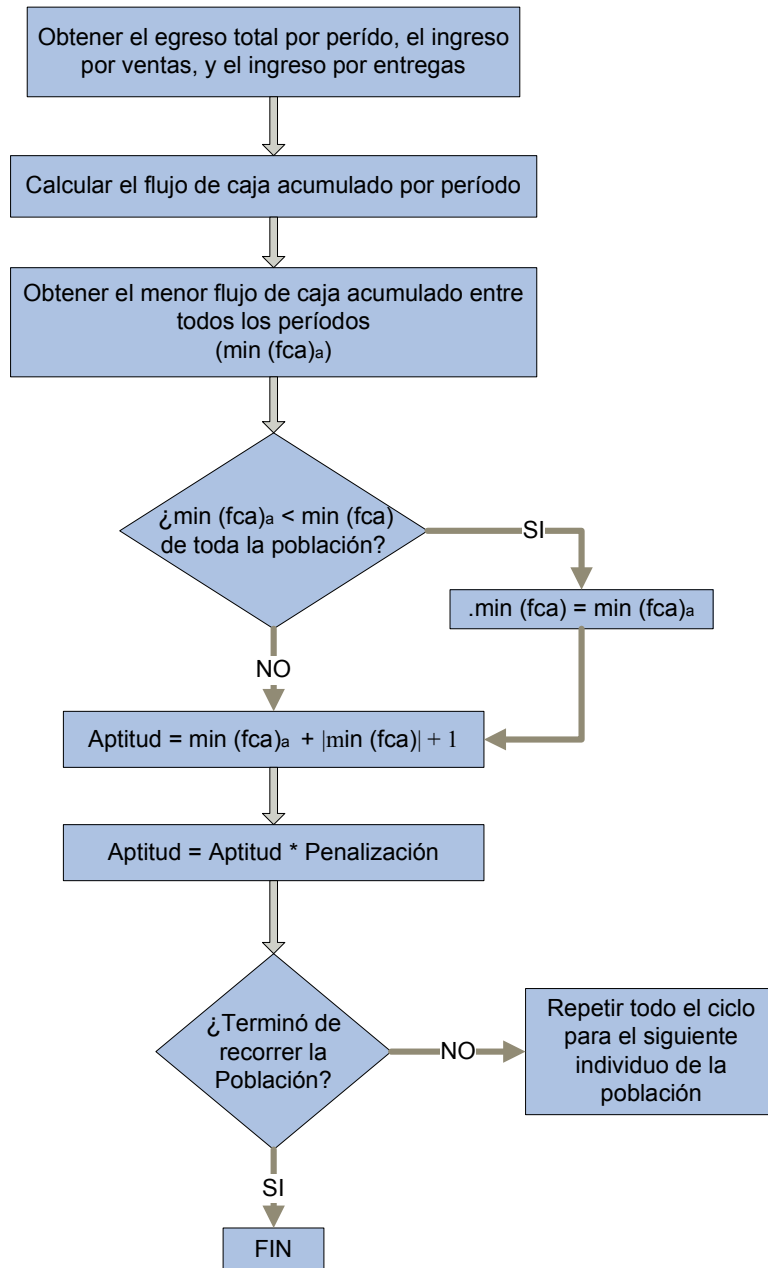


Diagrama 8. Diagrama de flujo de la función que calcula el valor de aptitud para FCSP



Tanto para la función objetivo MVPN como para FCSP se suma el mínimo flujo de caja acumulado, o el mínimo VPN, en valor absoluto dentro de la función de aptitud. Se hace de esta forma para evitar valores de aptitud negativos que afecten el funcionamiento de los operadores de selección; y se suma uno para que el valor de aptitud del individuo al que le corresponde el mínimo VPN, o flujo de caja acumulado, sea igual a uno y no a cero, lo que también afectaría los operadores de selección.

Al sumar en valor absoluto el valor del mínimo de la generación, se está garantizando que ningún valor de aptitud sea negativo, y al sumar uno se garantiza que ningún valor de aptitud sea cero. Cuando se evalúa el proceso evolutivo, lo que se hace normalmente es comparar el promedio del valor de aptitud entre las generaciones. Para este caso es necesario descontar al valor promedio de aptitud de cada generación, el valor del mínimo. El valor que se utiliza para evaluar el proceso evolutivo es el promedio del flujo de caja mínimo acumulado, o del VPN de todos los individuos que componen una generación.

3.7 ¿POR QUÉ UN ALGORITMO GENÉTICO?

La complejidad para seleccionar una buena programación de obra radica en la combinatoriedad. A partir de la información consignada en cada cromosoma se puede construir una programación de obra. Pero, ¿Cuántas soluciones posibles hay? La respuesta a esa pregunta refleja la magnitud del problema.

El cromosoma que se definió tiene dos características por actividad de construcción: duración y retraso. Si se supone una duración máxima de 10 días y mínima de 5 días por actividad, se tienen 6 posibles duraciones para cada actividad. Si se supone un retraso máximo de 3 días por actividad, se tienen 4 posibles valores de retraso para cada actividad. Para un proyecto de 10 actividades con estas características se tendrían $(6 \times 4)^{10} = 6.34 \times 10^{13}$ posibilidades de solución. Un computador que compare un millón de soluciones por segundo tardaría 2.04 años en compararlas todas. Tardaría más tiempo en encontrar la solución que el desarrollo del proyecto.

La forma general de la expresión para calcular la cantidad de soluciones posibles es la siguiente:

$$(D \times G)^A$$

Donde los parámetros son:

D = Tamaño del rango de duración de las actividades

G = Tamaño del rango de retraso de las actividades

A = Cantidad de actividades que componen el proyecto

Entre mayor sea la cantidad de actividades, más grande es el espacio solución. La complejidad del problema radica en la gran cantidad de combinaciones entre valores de duración y retraso de las actividades que pueden darse. Por eso los algoritmos genéticos constituyen una herramienta propicia para afrontar el problema.

4. DESCRIPCIÓN DE LA HERRAMIENTA

El producto de este trabajo de grado es una herramienta computacional programada en Visual Basic. La herramienta emplea un algoritmo genético para buscar la programación de obra que satisfaga las necesidades de dos escenarios bien definidos. Para cada programación de obra analizada, el programa construye el costo directo e indirecto del proyecto y lo ubica en el flujo de caja como un egreso. Los ingresos por su parte dependen de los ritmos de ventas y de entregas que el usuario ingresa.

El libro de Excel que contiene la herramienta está dividido en varias hojas, unas sirven como entrada de datos y otras hacen parte del algoritmo genético como tal.

4.1 INGRESO DE DATOS

El ingreso de datos consta del presupuesto de obra, el plan de ventas del proyecto y los parámetros de funcionamiento del algoritmo genético. A su vez el presupuesto se compone de insumos, análisis de precios unitarios, actividades y orden de precedencia entre actividades. El plan de ventas se compone del ritmo de ventas, y del de entregas.

4.1.1 Insumos. En esta hoja se ingresan todos los recursos materiales y de mano de obra disponibles en el proyecto:

Figura 17. Herramienta, Hoja de Insumos

	A	B	C	D	E
1	Codigo	Descripción	Unidad	Precio Unitario	Tipo
2	1	AGUA	LT	\$ 7	M
3	2	PLASTIFICANTE BV-40	KG	\$ 3.800	M
4	3	CEMENTO BLANCO	KG	\$ 680	M
5	4	ACERO DE REFUERZO	KG	\$ 1.330	M
6	8	IMPERMEABILIZANTE PARA MORTERO	KG	\$ 2.800	M
7	9	PUNTILLA	LB	\$ 1.200	M
8	12	ELEMENTOS DE FIJACION	GBL	\$ 100	M
9	13	LLAVE TERMINAL DE 1/2" GRICOL	UN	\$ 7.000	M
10	14	TUBERIA PVC PRESION 1/2"	ML	\$ 935	M
11	15	TUBERIA PVC PRESION 3/4"	ML	\$ 1.170	M
12	16	LAVAMANOS CHELSEA	UN	\$ 65.000	M
13	17	SANITARIO CHELSEA	UN	\$ 165.000	M
14	18	FORMALETA METALICA LOSA	M2	\$ 8.000	M
15	20	FORMALETA METALICA MUROS 1 CARA	M2	\$ 2.500	M
16	21	GRAVILLA	M3	\$ 24.000	M
17	22	ARENA	M3	\$ 20.000	M
18	23	TACO METALICO CORTO	DIA	\$ 100	M
19	25	TABLA 1"x8"x3.00 COMUN	UN	\$ 6.010	M
20	26	TELERA 2"x10"x3.00 SAJO	UN	\$ 13.200	M
21	27	LISTON 2"x2"x3.00 SAJO	UN	\$ 2.100	M
22	30	GUADUA SOBREBASA 4 VARAS	UN	\$ 1.500	M
23	32	TUBERIA PVC A. LLUVIAS 3"	ML	\$ 3.800	M
24	33	SOLDADURA 1/4 GALON	LIN	\$ 27.000	M

Cada insumo tiene que tener su valor unitario y debe estar identificado con M si es material o con MO si es un recurso de mano de obra.

4.1.2 Actividades. En esta hoja se ingresan todas las actividades que componen el proyecto:

Figura18. Herramienta, Hoja de Actividades

	A	B	C	D	E	F	G
1	Cod_Actividad	Descripción	Unidad	Cantidad	Durac Max	Durac Min	Retraso Max
2	1	REPLANTEO	M2	624	5	8	1
3	2	RAMADA	M2	75	8	10	1
4	3	NIVELACION LOTE	M3	100	10	15	2
5	4	INSTALACIONES PROVISIONALES	GL	1	2	3	1
6	5	CERRAMIENTO PROVISIONAL	GL	1	1	2	0
7	6	CONCRETO LOSA CIMENTACION TORRE 1	M3	50	1	2	3
8	7	CONCRETO LOSA CIMENTACION TORRE 2	M3	50	1	2	3
9	8	CONCRETO LOSA CIMENTACION TORRE 3	M3	50	1	2	3
10	9	CONCRETO LOSA CIMENTACION TORRE 4	M3	50	1	2	3
11	10	CONCRETO LOSA CIMENTACION TORRE 5	M3	50	1	2	3
12	11	EXCAVACION MANUAL CIMIENTOS TORRE 1	M3	38	1	2	1
13	12	EXCAVACION MANUAL CIMIENTOS TORRE 2	M3	38	1	2	1
14	13	EXCAVACION MANUAL CIMIENTOS TORRE 3	M3	38	1	2	1
15	14	EXCAVACION MANUAL CIMIENTOS TORRE 4	M3	38	1	2	1
16	15	EXCAVACION MANUAL CIMIENTOS TORRE 5	M3	38	1	2	1
17	16	RELLENOS DE EXCAVACION CIMIENTOS TORRE 1	M3	14	1	2	1
18	17	RELLENOS DE EXCAVACION CIMIENTOS TORRE 2	M3	14	1	2	1
19	18	RELLENOS DE EXCAVACION CIMIENTOS TORRE 3	M3	14	1	2	1
20	19	RELLENOS DE EXCAVACION CIMIENTOS TORRE 4	M3	14	1	2	1
21	20	RELLENOS DE EXCAVACION CIMIENTOS TORRE 5	M3	14	1	2	1
22	21	RETIRO DE SOBRESANTES CIMIENTOS TORRE 1	M3	35	1	2	1
23	22	RETIRO DE SOBRESANTES CIMIENTOS TORRE 2	M3	35	1	2	0
24	23	RETIRO DE SOBRESANTES CIMIENTOS TORRE 3	M3	35	1	2	0
25	24	RETIRO DE SOBRESANTES CIMIENTOS TORRE 4	M3	35	1	2	0

Es importante notar que cada actividad tiene un código, una cantidad de ejecución, una duración máxima y mínima, y un retraso máximo. En un principio no se tuvo en cuenta el retraso máximo, sin embargo, luego de ensayar el programa varias veces se notó que al restringir el retraso de cada actividad se disminuye el tamaño del espacio solución, lo cual hace más eficiente el uso del algoritmo. El espacio solución se disminuye pues existe un rango preestablecido para el retraso de cada actividad, no hay necesidad de explorar soluciones con retrasos por fuera de ese rango que seguramente van a tener penalizaciones por incumplimiento en los plazos de entrega.

4.1.3 Análisis de precios unitarios. En esta hoja se ingresan los insumos requeridos por cada actividad y la cantidad por unidad de ejecución de la actividad:

Figura 19. Herramienta, Hoja Unitarios

	A	B	C	D
1	Cod_Actividad	Cod_Insumo	Cantidad	Rendimiento
2	1	55	0,600	
3	1	27	0,100	
4	1	9	0,020	
5	1	300		0,100
6	2	245	1,000	
7	3	300		0,575
8	4	285	1,000	
9	5	286	1,000	
10	6	25	1,000	
11	6	303	1,000	
12	6	300		5,556
13	7	25	1,000	
14	7	303	1,000	
15	7	300		5,556
16	8	25	1,000	
17	8	303	1,000	
18	8	300		5,556
19	9	25	1,000	

En este ejemplo la actividad 1 requiere de los materiales con código 55, 27 y 9, y de la mano de obra con código 300. Cuando el insumo es un material, se introduce la cantidad de

insumo necesaria para completar una unidad de la actividad; si es mano de obra se introduce su rendimiento, es decir, cuantas unidades de la actividad realiza en una hora.

4.1.4 Precedencias. Debe ingresarse para cada actividad sus predecesoras:

Figura 20. Herramienta, Hoja Precedencias

	A	B	C	D	E	F
1	Cod_Actividad	Predecesoras				
2	1					
3	2	3				
4	3	1				
5	4	3				
6	5	1				
7	6	31	463	488	533	583
8	7	32	468	493	538	588
9	8	33	473	498	543	593
10	9	34	478	503	548	598
11	10	35	483	508	553	603
12	11	4				
13	12	4				
14	13	4				
15	14	4				
16	15	4				
17	16	36				
18	17	37				
19	18	38				
20	19	39				
21	20	40				
22	21	16				
23	22	17				
24	23	18				

Por ejemplo la actividad 6 tiene como predecesoras la 31, 463, 488, 533, 583.

4.1.5 Ingreso. En esta hoja se ingresa el plan de ventas, el plan de entregas, y los costos indirectos del proyecto:

Figura 21. Herramienta, Hoja Ingreso

CONTROL DE INGRESOS		RITMO DE VENTAS			
		Porcentaje de Cuota Inicial =		30%	
		Periodos de Plazo para Pago de Cuota Inicial =		5	
Período	Cantidad de Unidades Vendidas	Valor de Venta	Ingreso al Momento de la Venta		
-1	2	\$ 98.080.122,91	\$ 58.848.073,75		
1	2	\$ 98.080.122,91	\$ 58.848.073,75		
2	1	\$ 98.080.122,91	\$ 29.424.036,87		
3	2	\$ 98.080.122,91	\$ 58.848.073,75		
4	2	\$ 98.080.122,91	\$ 58.848.073,75		
5	1	\$ 98.080.122,91	\$ 29.424.036,87		
\$ 7.954.000,00	Costo de Administración Mensual				
\$ 28.735.706,00	Honorarios de los Diseños				
\$ 5.500.000,00	Licencia de Construcción				
6%	Tasa de Descuento				

En esta parte deben ingresarse costos tales como la administración mensual, los honorarios de los diseños arquitectónicos, estructurales, hidráulicos, de suelos etc. También se ingresa el plan de ventas estimado, que incluye el porcentaje de cuota inicial, el plazo de pago de la cuota inicial, el periodo en el cual ocurre la venta y la cantidad de unidades vendidas en cada periodo.

Muchos proyectos tienen una etapa de pre-venta durante la cual si bien hay ingresos de dinero, no hay egresos causados por la construcción misma de la edificación. Por eso en el ejemplo mostrado se venden dos unidades en el período -1. La construcción del proyecto siempre inicia en el período 1, y el flujo de caja del proyecto se construye a partir del período 1, pero es necesario tener en cuenta que al iniciar el periodo 1 puede haber dinero proveniente de las pre-ventas.

También en esta hoja se ingresa el plan de entregas y los costos indirectos asociados a la entrega de las unidades de vivienda:

Figura 22. Herramienta, Hoja Ingreso

PLAZO DE ENTREGAS						
Porcentaje al momento de la entrega =		70%				
Codigo de Actividad Limite	Día Limite de Entrega	Valor de Venta de la(s) Unidades Entregadas	Ingreso al Momento de la Entrega	Valor del Terreno de la(s) Unidades Entregadas	Costos Notariales de la(s) Unidades Entregadas	Pago de Comisiones de la(s) Unidades Entregadas
6	150	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
11	150	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
16	180	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
21	180	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
26	210	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
31	210	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
36	240	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
41	240	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
46	270	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00
51	270	\$ 98.080.122,91	\$ 68.656.086,04	\$ 20.000.000,00	\$ 2.980.000,00	\$ 3.000.000,00

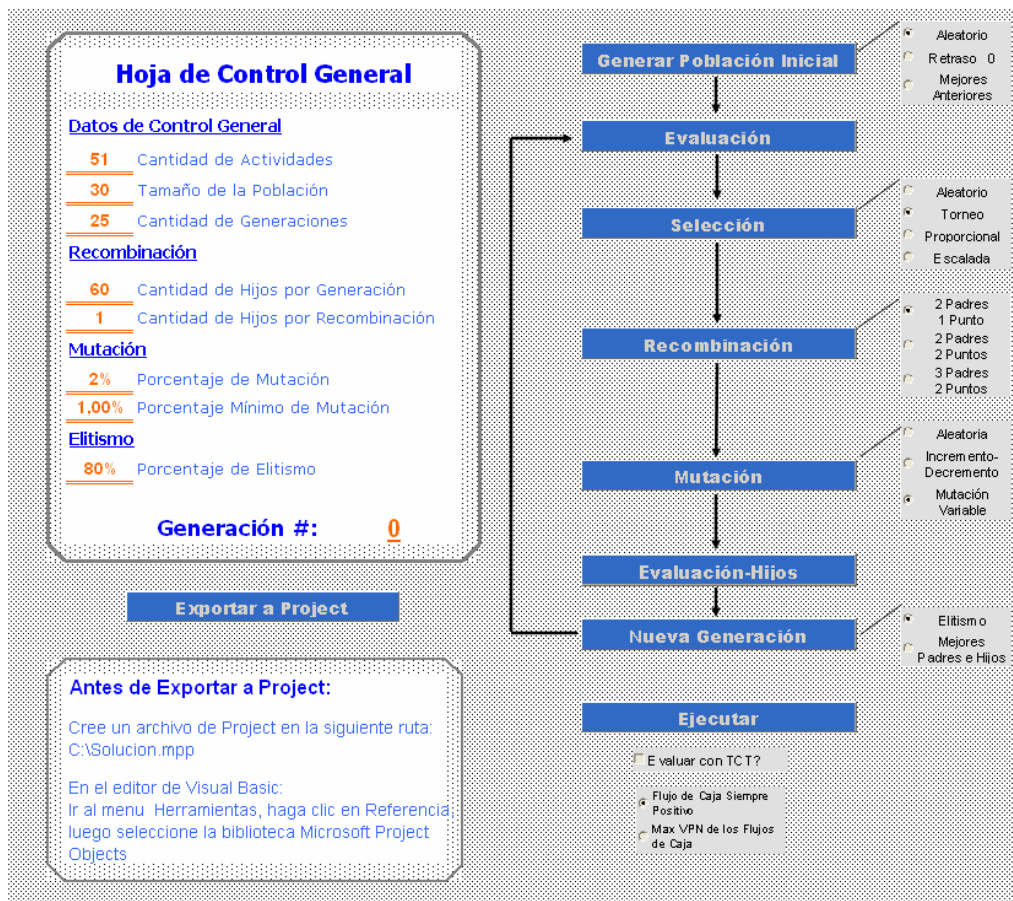
Existe un plazo máximo para entregar las unidades de vivienda a los compradores. Este plazo máximo no puede ser arbitrario pues si es muy largo los clientes preferirán comprar en otros proyectos, y si es muy corto no es posible hacer la construcción en tan poco tiempo.

Existen actividades con las cuales se finaliza la construcción de una o varias unidades. Con dichas actividades se controla el ritmo de entregas del proyecto. En el ejemplo, la actividad 6 determina el final de una unidad de vivienda. El día final de dicha actividad no puede ser superior al día 150, de ser así la solución sería penalizada pues no cumple con los requerimientos del mercado. Si el día final de la actividad 6 es menor o igual a 150, entonces la solución es válida y el ingreso de dinero restante ocurrirá no necesariamente el día 150 sino el día en que la unidad esté lista, es decir, el día en que la actividad 6 se haya finalizado.

Al momento de hacer la entrega del inmueble al propietario también ocurren otros costos como: el pago de la proporción de terreno correspondiente a la unidad entregada, los gastos notariales y los pagos a terceros de comisiones por ventas.

4.1.6 Control. En esta hoja se controla todo lo concerniente al algoritmo genético:

Figura 23. Herramienta, Hoja Control



La hoja contiene el esquema lógico del algoritmo genético y también los parámetros de funcionamiento del mismo. En esta hoja se ingresan dichos parámetros y se escogen los diferentes operadores disponibles (los operadores genéticos empleados están explicados en el capítulo 3). Al presionar el botón ejecutar, el algoritmo se ejecuta un número de veces igual al número de generaciones que el usuario defina. Si se presiona botón por botón, el usuario puede ejecutar el algoritmo paso a paso para una generación.

En esta hoja también se escoge la función objetivo (ver sección 3.6) a emplear y se decide el uso de TCT.

El botón Exportar a Project permite al usuario obtener la programación de obra del mejor individuo encontrado, y reportado en la Hoja Reporte (ver sección 4.2.5), en el programa Microsoft Project. Es importante seguir las instrucciones previas a la exportación a Microsoft Project que se han consignado en la Hoja Control, de lo contrario aparecerán errores al ejecutar la aplicación.

Una vez se ha ejecutado la exportación a Microsoft Project, aparece la actividad “Inicio”. Dicha actividad ha sido incluida en la programación de obra para poder desplazar el inicio del proyecto. Project inicializa el proyecto en el día actual, para desplazar el proyecto hacia la fecha real en que se tiene previsto empiece la construcción, solo es necesario desplazar la fecha de inicio de la actividad “Inicio”.

4.3 ALGORITMO GENÉTICO

Las hojas de funcionamiento del algoritmo no deben ser modificadas directamente por el usuario, sino a través de la hoja de control.

4.3.1 Población. La hoja población contiene los padres de cada generación. En la primera generación estos son creados según la opción que el usuario escoja. En las siguientes generaciones esta hoja contiene a los mejores individuos de la generación anterior que fueron seleccionados mediante elitismo o mejores padres e hijos (ver sección 3.5).

4.3.2 Población Intermedia. La población intermedia está constituida por los hijos que resultan de la recombinación entre parejas o tríos de padres según el operador de recombinación seleccionado (ver sección 3.3). La primera columna de esta hoja contiene el listado de los padres seleccionados (ver sección 3.2), a partir de la segunda columna se encuentran los hijos generados con los padres listados en la primera columna.

4.3.3 Población Mejores. En esta hoja se organizan los individuos que son seleccionados para ser los padres de la siguiente generación.

4.3.4 Resultados Previos. En este espacio se almacena el mejor resultado obtenido luego de terminar de ejecutar el algoritmo. Pueden almacenarse máximo doscientos cincuenta y seis resultados previos. Estos resultados pueden ser utilizados nuevamente como población inicial de una nueva ejecución del algoritmo, seleccionando la opción *Mejores Anteriores* en la hoja de Control.

4.3.5 Reporte. En la hoja de reporte se registran los valores promedios de aptitud generación a generación y el mejor individuo encontrado. En esta hoja también quedan registrados los parámetros bajo los cuales se ejecutó el algoritmo.

El valor realmente importante generación a generación no es el promedio del valor de aptitud como tal. El registro sobre la mejoría en la población debe hacerse teniendo en cuenta el valor mínimo del VPN o del flujo de caja acumulado en cada generación. El valor de aptitud entre una generación y otra no es consistente pues en una generación con un individuo muy malo los valores de aptitud son mayores que en una generación con individuos muy buenos pero más parejos. Esto se debe a que en la función de aptitud se suma siempre, al VPN o al menor flujo de caja acumulado de cada individuo, el valor mínimo de toda la población en valor absoluto para garantizar que todas las aptitudes sean positivas. De esta forma el valor promedio de aptitud está afectado también por el mínimo de la población. Para observar en términos reales como está mejorando la población, lo que se hace es restar al promedio del valor de aptitud, el valor mínimo de VPN o flujo de caja acumulado de la población si éste es positivo, o sumarlo si es negativo. De esta forma se obtiene el valor real del promedio del VPN o del menor flujo de caja para cada generación (ver sección 3.6).

5. DETERMINACIÓN DE LOS PARÁMETROS DE FUNCIONAMIENTO DEL ALGORITMO GENÉTICO

La calibración de la herramienta consiste en probar varias combinaciones de operadores genéticos con parámetros de funcionamiento iguales. El objetivo es encontrar la combinación de operadores con la que se logra el mejor funcionamiento de la herramienta. El proyecto sobre el cual se hizo todo el análisis consta de 51 actividades.

La calibración se hizo para las dos funciones objetivo (FCSP y MVPN) de la misma forma. Lo primero fue comparar los métodos de selección, luego los operadores de recombinación y finalmente entre los tipos de mutación disponibles.

Los siguientes parámetros se dejaron fijos para todas las comparaciones:

Tamaño de la población = 30

Cantidad de generaciones = 25

Cantidad de hijos por generación = 60

Cantidad de hijos por recombinación = 1

Porcentaje de Mutación = 2% para FCSP y 15% para MVPN

Porcentaje mínimo de mutación = 1%

Elitismo = 80%

Estos parámetros se dejaron fijos para garantizar que todos los operadores se comparen en igualdad de condiciones.

El esquema bajo el cual se hizo la comparación de los operadores es el siguiente:

Figura 24. Esquema de decisión para la comparación entre operadores genéticos

Comparación entre Métodos de Selección:

Torneo:	1. Torneo - Mejores padres e hijos	Método de Selección Escogido
	2. Torneo - Elitismo	
Proporcional:	3. Proporcional - Mejores padres e hijos	
	4. Proporcional - Elitismo	
Escalada:	5. Escalada - Mejores padres e hijos	
	6. Escalada - Elitismo	

Comparación entre Operadores de Recombinación:

<i>Método de Selección Escogido</i>	7. 2 Padres 1 Punto	Operador de Recombinación Escogido
	8. 2 Padres 2 Puntos	
	9. 3 Padres 2 Puntos	

Comparación entre Operadores de Mutación:

<i>Operador de Recombinación Escogido</i>	10. Aleatoria	Operador de Mutación Escogido
	11. Incremento - Decremento	
	12. Variable	

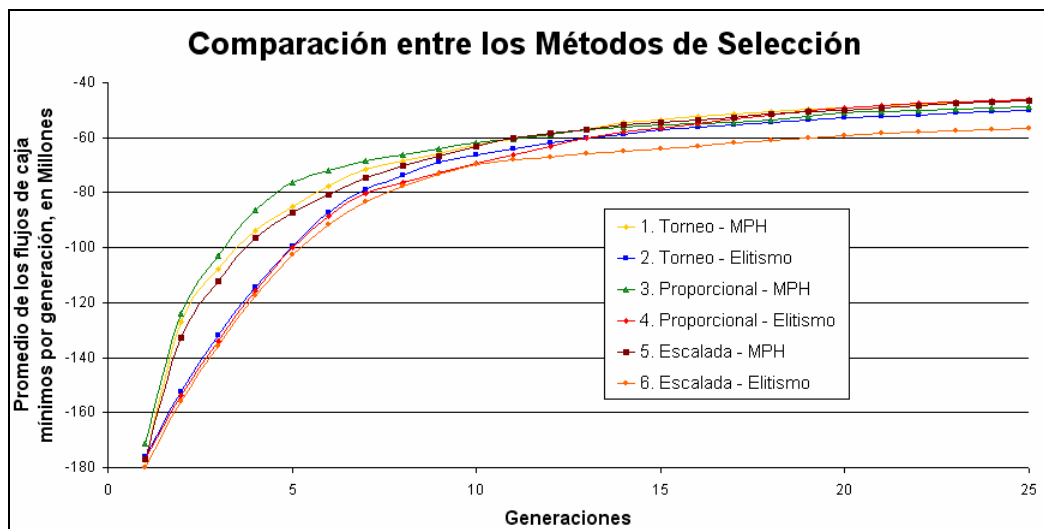
Este esquema indica las diferentes combinaciones entre operadores con las cuales se debe ejecutar el algoritmo para seleccionar los operadores que mejores resultados ofrecen en el funcionamiento de la herramienta. La determinación de los mejores operadores se hace de la misma forma tanto para FCSP como para MVPN. Se realizan tres ejecuciones de la herramienta para cada una de las combinaciones planteadas en el esquema de decisión (Figura 24). Es decir que se realizan en total treinta y seis ejecuciones de la herramienta de la siguiente forma: dieciocho ejecuciones para comparar las seis combinaciones del método de selección, nueve ejecuciones para comparar los tres operadores de recombinación, y nueve ejecuciones para comparar los tres operadores de mutación. Finalmente la escogencia de los mejores operadores para FCSP y MVPN exige setenta y dos ejecuciones de la herramienta (treinta y seis por cada función objetivo).

Durante la comparación entre los métodos de selección se utilizan siempre los operadores recombinación 2 Padres 1 Punto y Mutación Aleatoria. La comparación entre operadores de recombinación se hace siempre con Mutación Aleatoria.

5.1 DETERMINACIÓN DE LOS MEJORES OPERADORES PARA EL USO DE FCSP

5.1.1 Determinación del método de selección. Se realizaron las tres ejecuciones por combinación, establecidas en el esquema de decisión. Con el resultado de cada una de las tres ejecuciones se establece un promedio, que se emplea para evaluar cada uno de los métodos de selección. El resultado está consignado en la siguiente gráfica, donde se muestra el proceso evolutivo obtenido con cada una de las combinaciones establecidas para la determinación del mejor método de selección:

Gráfica 4. Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de FCSP

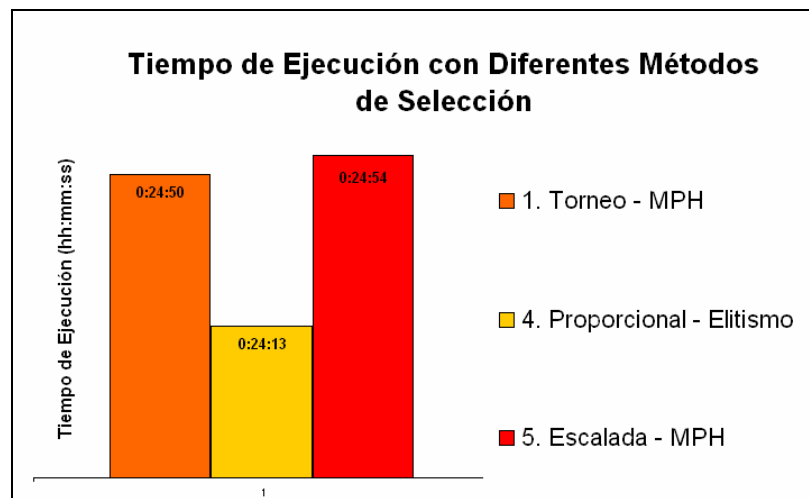


En la gráfica anterior se observa que los mejores resultados del proceso evolutivo se obtuvieron con las combinaciones 1 (Torneo – MPH), 4 (Proporcional – Elitismo) y 5 (Escalada – MPH). El comportamiento de las combinaciones 1 y 5 es muy similar, se

incrementa rápidamente el promedio de los flujos de caja mínimos³² en las primeras generaciones. Al final de la curva, el valor de aptitud varía muy poco entre una generación y otra. En cambio la combinación 4 se comporta diferente. Si bien la pendiente de la curva disminuye hacia el final, la pendiente es mayor lo que indica un cambio entre el valor de aptitud de una generación y otra mayor al de las combinaciones 1 y 5. La tendencia de la curva 4 en las últimas generaciones indica una mejoría más rápida del valor de aptitud promedio, por lo cual es más deseable el comportamiento de esta combinación, sobre las otras dos en consideración.

Otro factor importante para definir la mejor combinación es el tiempo de ejecución del algoritmo. La Gráfica 5 muestra la diferencia entre los tiempos de ejecución para las combinaciones 1, 4 y 5.

Gráfica 5. Tiempos de ejecución del algoritmo con diferentes métodos de selección para el uso de FCSP



El tiempo de ejecución para la combinación 4 es menor que para las combinaciones 1 y 5. Quiere decir entonces que se obtuvo con la combinación 4 un resultado más deseable y con un menor tiempo de ejecución. La combinación 4 (Proporcional – Elitismo), se constituye

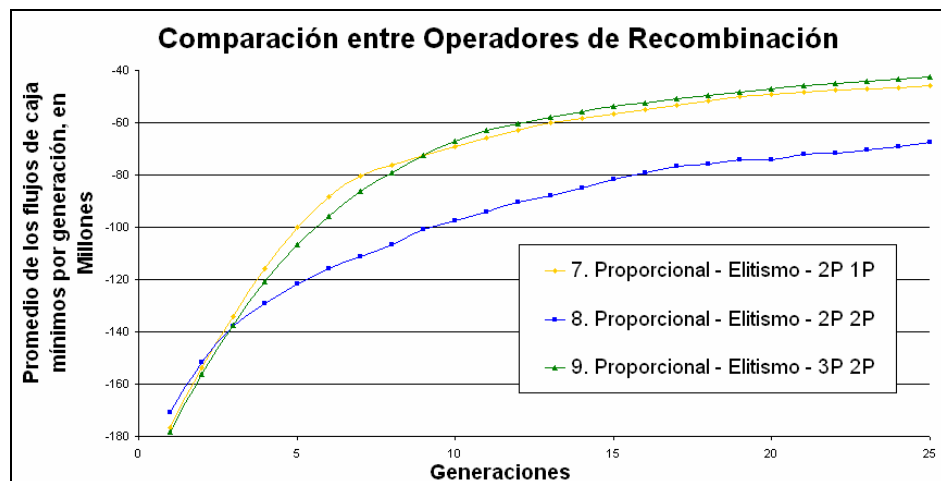
³² Es indiferente el uso de las expresiones: flujos de caja mínimos o aptitud, en este capítulo.

como el *Método de Selección Escogido*, según lo planteado en el esquema de decisión (Figura 24).

5.1.2 Determinación del operador de recombinación. La comparación entre los operadores de recombinación se hizo con el promedio de tres ejecuciones de la herramienta por operador. El esquema de decisión indica que el método de selección a emplear, siempre debe ser el escogido en la fase anterior, para este caso el 4 (Proporcional – Elitismo).

La Gráfica 6 muestra los procesos evolutivos obtenidos luego de ejecutar la herramienta con cada uno de los métodos de recombinación:

Gráfica 6. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de FCSP



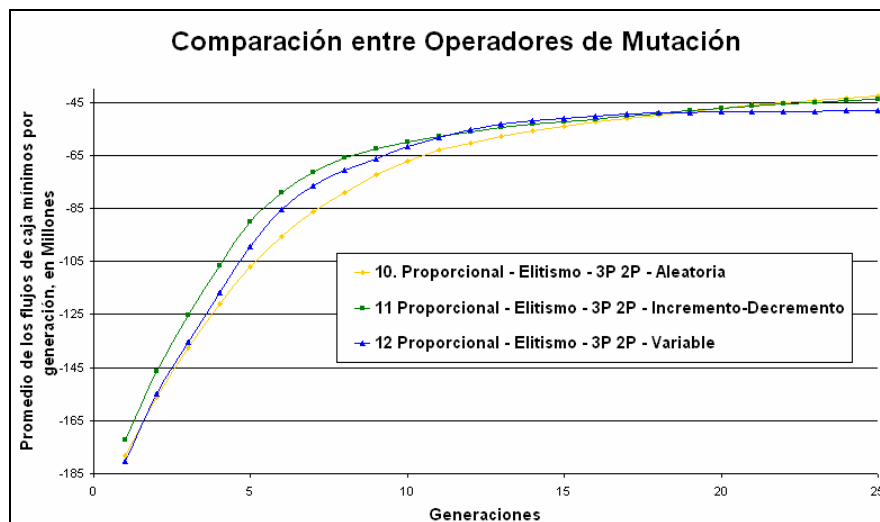
El mejor comportamiento del proceso evolutivo se obtiene con los operadores 2 Padres 1 Punto (combinación 7) y 3 Padres 2 Puntos (combinación 9). Resulta mejor la combinación 9 por dos motivos: El primero es que inicia con un promedio de aptitud en la primera generación más bajo que el de la combinación 7 y aún así se logra un promedio de aptitud mayor a partir de la décima generación. El segundo motivo para seleccionar el operador de recombinación 3 Padres 2 Puntos es que la pendiente de la curva en las últimas generaciones es mayor, dicha tendencia indica que el valor de aptitud mejora con mayor

dinamismo que el ofrecido por las combinaciones 7 y 8. El operador de selección escogido es 3 Padres 2 Puntos pues se hace evidente que su uso es más provechoso según lo observado en la Gráfica 6.

5.1.3 Determinación del operador de mutación. Anterior a esta etapa se ha determinado como método de selección la combinación Proporcional – Torneo y el operador de recombinación escogido es 3 Padres 2 Puntos. Dichos operadores se combinan con cada uno de los tres operadores de mutación. Se ejecuta la herramienta tres veces por cada combinación. Con el promedio de las tres ejecuciones por combinación se hace la comparación entre los operadores de mutación.

Tal comparación se hace a partir del comportamiento del proceso evolutivo obtenido con el uso de cada uno de ellos. La Gráfica 7 muestra dicho comportamiento:

Gráfica 7. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de FCSP



La mutación aleatoria empleada en la combinación 10 refleja el mejor comportamiento del proceso evolutivo. En las primeras generaciones el valor de aptitud no tiene un incremento tan rápido como sucede con las combinaciones 11 (Incremento-Decremento) y 12

(Mutación Variable). Pero, en las últimas generaciones el valor de aptitud crece de tal forma (con la mutación aleatoria) que supera el resultado obtenido con los otros dos métodos de mutación, lo cual indica que el proceso de búsqueda es más efectivo. La mayor efectividad en la búsqueda se obtuvo con la mutación aleatoria, por este motivo es el *Operador de Mutación Escogido*.

Los operadores escogidos para el uso de FCSP son:

Método de Selección: Proporcional – Elitismo

Operador de Recombinación: 3 Padres 2 Puntos de Cruce

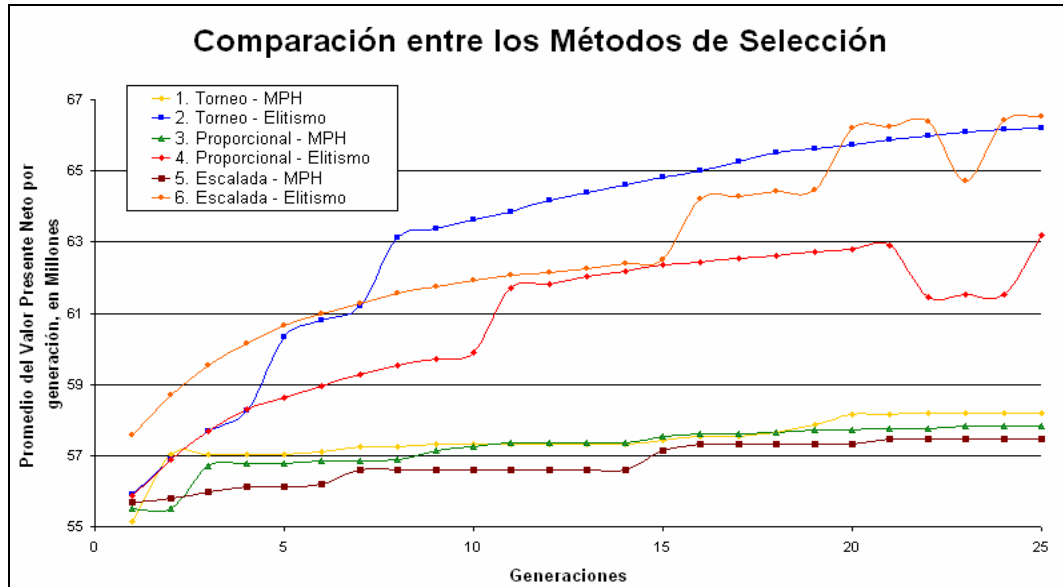
Operador de Mutación: Mutación Aleatoria

5.2 DETERMINACIÓN DE LOS MEJORES OPERADORES PARA EL USO DE MVPN

5.2.1 Determinación del método de selección. El proceso de determinación del mejor método de selección para el uso de MVPN se hizo de la misma forma que para el uso de FCSP. Se realizaron igualmente tres ejecuciones por combinación. La gráfica 8 muestra el comportamiento durante el proceso evolutivo del valor de aptitud³³ promedio en cada generación:

³³ Es indiferente el uso de las expresiones: valor presente neto o aptitud, en este capítulo.

Gráfica 8. Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de MVPN



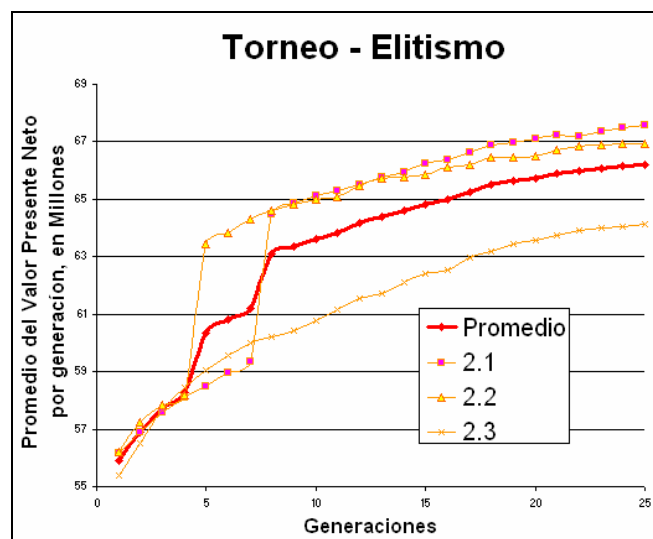
En las combinaciones 2 y 6 se evidencia el mejor comportamiento del algoritmo. El elitismo parece tener muy buenas implicaciones sobre el funcionamiento de la herramienta pues en este caso el mejor comportamiento del proceso evolutivo se obtiene con dos combinaciones que involucran el elitismo; adicionalmente el *Método de Selección Escogido* para el uso de FCSP también involucra el elitismo.

El comportamiento de la combinación 2 es más deseable que el de la combinación 6, se observa mayor consistencia hacia el final de la curva. Esta situación es deseable pues indica una mejor configuración del proceso evolutivo en cuanto a que la evolución si bien es más lenta, no presenta sobresaltos que evidencian cierta aleatoriedad en el proceso. La combinación 6 en las últimas generaciones tiene un comportamiento poco predecible presumiblemente por causa de la mutación. La selección escalada, involucrada en la combinación 6, incrementa la probabilidad de los individuos débiles para ser seleccionados. Puede suceder que la mutación altere algunos individuos de la población en detrimento de su aptitud. La selección escalada abre la posibilidad para que dichos individuos sean

seleccionados más fácilmente y por eso el comportamiento en las últimas generaciones se hace poco predecible. Por el contrario, con la combinación 2 no se evidencia ese problema.

La siguiente gráfica muestra el comportamiento de cada una de las tres ejecuciones realizadas para la combinación 2:

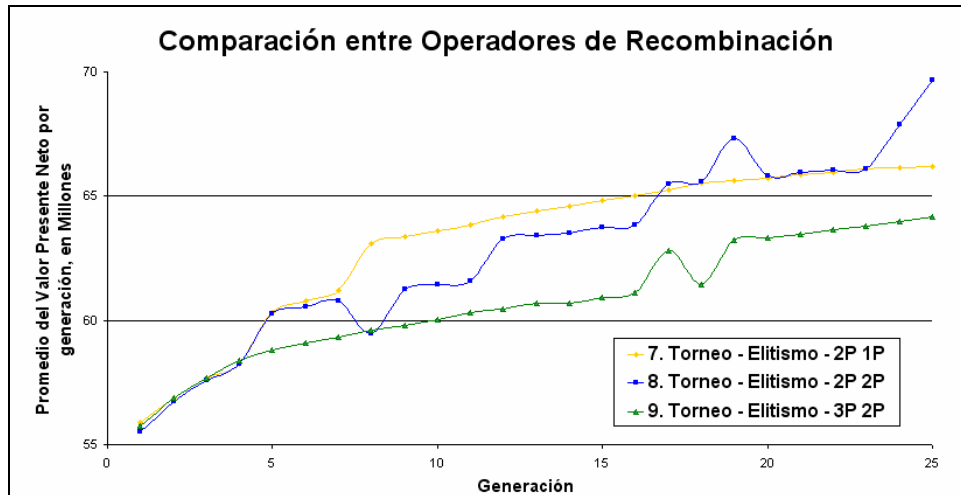
Gráfica 9. Proceso evolutivo obtenido con la combinación 2 (Torneo – Elitismo), para el uso de MVPN



Se observa claramente en la gráfica 9 que el comportamiento de la combinación 2 es predecible en las últimas generaciones. Esta característica confirma que el *Método de Selección Escogido* para el uso de MVPN sea la combinación 2, Torneo – Elitismo.

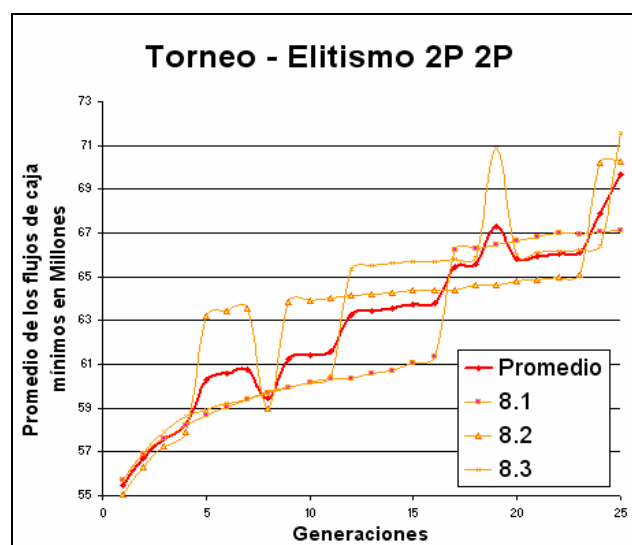
5.2.2 Determinación del operador de recombinación. Se ejecutó la herramienta combinando Torneo - Elitismo con los tres operadores de recombinación. El resultado es el siguiente:

Gráfica 10. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de MVPN



A primera vista parecería que la combinación 8 (2 Padres 2 Puntos) ofrece el mejor resultado. Es necesario observar el resultado de cada una de las tres ejecuciones realizadas con la combinación 8 (Torneo – Elitismo – 2P 2P), para confirmar que no es tan deseable su comportamiento:

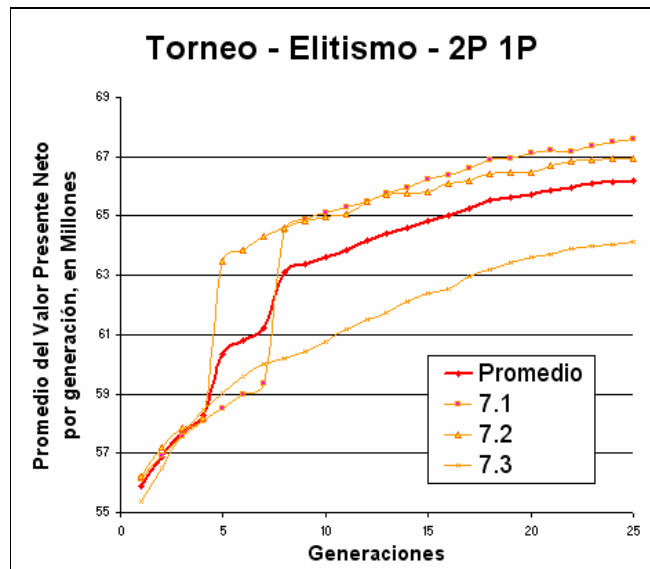
Gráfica 11. Proceso evolutivo obtenido con la combinación 8 (Torneo – Elitismo – 2P 2P), para el uso de MVPN



La gráfica 11 permite observar con claridad que el proceso evolutivo obtenido con esta combinación no es muy consistente. No existe correspondencia entre el comportamiento de una curva y otra, la falta de coherencia en el comportamiento de las tres ejecuciones le resta confiabilidad a esta combinación de operadores.

La combinación 7 (Torneo – Elitismo – 2 Padres 1 Punto) aunque no alcanzó los valores de aptitud de la combinación 8, tiene un comportamiento más confiable:

Gráfica 12. Proceso evolutivo obtenido con la combinación 7 (Torneo – Elitismo – 2P 1P), para el uso de MVPN

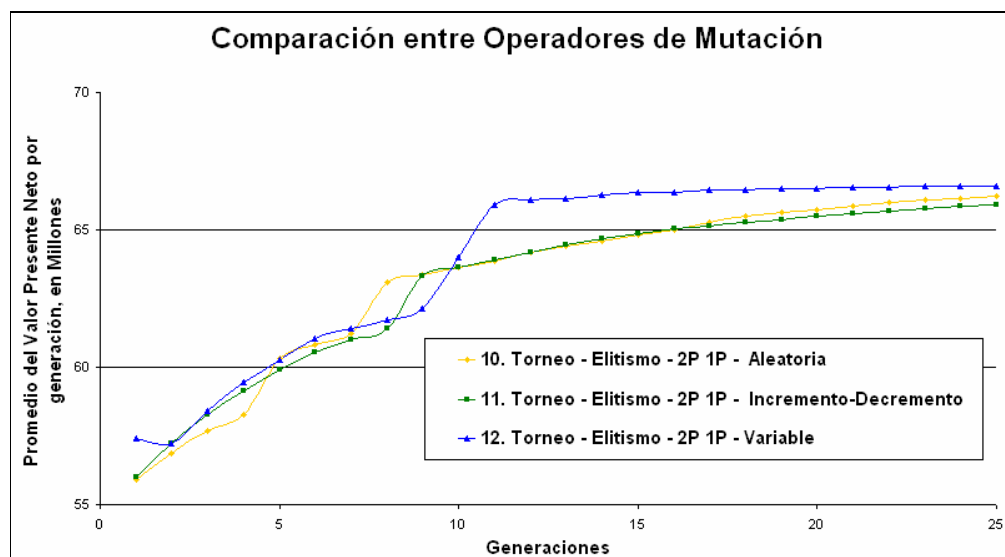


Los resultados obtenidos con esta combinación tienen gran coherencia entre ellos. El proceso evolutivo es más predecible y eso le da mayor confiabilidad. Por eso el *Operador de Recombinación Escogido* es 2 Padres 1 Punto.

5.2.3 Determinación del operador de mutación. El comportamiento de las tres combinaciones es bien parecido. Con la combinación 12 (Mutación Variable) se obtienen los mayores valores de aptitud, pero la tendencia de la curva en las últimas generaciones no indica un aumento significativo en el valor promedio de aptitud. La combinación 10

(Mutación Aleatoria) no muestra un incremento tan veloz en el valor promedio de aptitud pero su tendencia en las últimas generaciones indica un mayor crecimiento del valor de aptitud entre una generación y otra.

Gráfica 13. Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de MVPN



El *Operador de Mutación Escogido* es Mutación Aleatoria.

Los operadores escogidos para el uso de MVPN son:

Método de Selección: Torneo – Elitismo

Operador de Recombinación: 2 Padres 1 Punto de Cruce

Operador de Mutación: Mutación Aleatoria

La calibración del algoritmo genético, para el uso de cada función objetivo, es indispensable. Sólo de esta forma se puede encontrar la combinación de operadores genéticos con la que se logra el mejor funcionamiento del algoritmo. Adicionalmente, es importante notar que los operadores escogidos fueron diferentes para MVPN y para FCSP.

6. ESTUDIO DE CASO

6.1 DESCRIPCIÓN DEL PROYECTO

La herramienta fue probada para la planeación de un proyecto de diez casas, compuesto por cincuenta y un actividades. El nombre del proyecto es *Portal del Bosque*, y se desarrolló en Manizales en el año 2000.

Cada una de las actividades que componen el proyecto, tiene una duración máxima y mínima, y un retraso máximo. Las siguientes son las actividades que componen el proyecto, con sus condiciones de duración y retraso:

Cuadro 8. Contiene las actividades que componen el proyecto *Portal del Bosque*.

Cod Actividad	Descripción	Duración Máxima	Duración Mínima	Retraso Máximo
1	Preliminares Obra	7	1	30
2	Urbanismo Casa1	40	30	5
3	Preliminares Casa1	8	6	3
4	Estructura Casa 1	30	20	3
5	Cubierta Casa 1	7	5	5
6	Acabados Casa 1	25	20	2
7	Urbanismo Casa2	40	30	5
8	Preliminares Casa2	8	6	3
9	Estructura Casa2	30	20	3
10	Cubierta Casa 2	7	5	5
11	Acabados Casa 2	25	20	2
12	Urbanismo Casa3	40	30	35
13	Preliminares Casa3	8	6	3
14	Estructura Casa3	30	20	3
15	Cubierta Casa 3	7	5	5
16	Acabados Casa 3	25	20	2
17	Urbanismo Casa4	40	30	35
18	Preliminares Casa4	8	6	3
19	Estructura Casa4	30	20	3
20	Cubierta Casa 4	7	5	5

21	Acabados Casa 4	25	20	2
22	Urbanismo Casa5	40	30	65
23	Preliminares Casa5	8	6	3
24	Estructura Casa5	30	20	3
25	Cubierta Casa 5	7	5	5
26	Acabados Casa 5	25	20	2
27	Urbanismo Casa6	40	30	65
28	Preliminares Casa6	8	6	3
29	Estructura Casa6	30	20	3
30	Cubierta Casa 6	7	5	5
31	Acabados Casa 6	25	20	2
32	Urbanismo Casa7	40	30	95
33	Preliminares Casa7	8	6	3
34	Estructura Casa7	30	20	3
35	Cubierta Casa 7	7	5	5
36	Acabados Casa 7	25	20	2
37	Urbanismo Casa8	40	30	95
38	Preliminares Casa8	8	6	3
39	Estructura Casa8	30	20	3
40	Cubierta Casa 8	7	5	5
41	Acabados Casa 8	25	20	2
42	Urbanismo Casa9	40	30	125
43	Preliminares Casa9	8	6	3
44	Estructura Casa9	30	20	3
45	Cubierta Casa 9	7	5	5
46	Acabados Casa 9	25	20	2
47	Urbanismo Casa10	40	30	125
48	Preliminares Casa10	8	6	3
49	Estructura Casa10	30	20	3
50	Cubierta Casa 10	7	5	5
51	Acabados Casa 10	25	20	2

La última actividad en la construcción de cada casa son los acabados. Una vez se termina de ejecutar dicha actividad, la unidad de vivienda puede ser entregada al cliente.

El precio de venta de cada unidad es: \$ 98.080.122 y las condiciones de pago son las siguientes:

- Cuota Inicial: \$ 29.424.036 (30% del precio de venta)
- Plazo de pago de la cuota inicial: 5 meses
- Pago al momento de la entrega de la unidad de vivienda: \$ 68.656.086 (70% del precio de venta)

Se tiene pronosticado vender las casas en seis meses de la siguiente forma:

Cuadro 9. Contiene el plan de ventas del proyecto *Portal del Bosque*

Período	Cantidad de Unidades Vendidas	Valor de Venta	Ingreso al Momento de la Venta
-1	2	\$ 98.080.122,91	\$ 58.848.073,75
1	2	\$ 98.080.122,91	\$ 58.848.073,75
2	1	\$ 98.080.122,91	\$ 29.424.036,87
3	2	\$ 98.080.122,91	\$ 58.848.073,75
4	2	\$ 98.080.122,91	\$ 58.848.073,75
5	1	\$ 98.080.122,91	\$ 29.424.036,87

Las ventas inician antes que la obra. Se pretenden vender dos unidades en el periodo anterior al del inicio de la construcción. Cada periodo es igual a un mes de 30 días. El plazo de pago de la cuota inicial es de cinco meses, con lo cual las cuotas iniciales de las dos unidades vendidas en el período -1 se terminarán de pagar en el período 4; y la cuota inicial de la unidad vendida en el período 5 se terminará de pagar en el período 9.

El plazo máximo para la entrega de cada unidad es de 6 meses a partir de la venta. Por ejemplo las dos casas vendidas en el período -1, podrán entregarse máximo en el período 5; y la unidad vendida en el quinto mes podrá entregarse máximo en el décimo mes. Por eso el plan de entregas es el siguiente:

Cuadro 10. Contiene los plazos de entrega del proyecto *Portal del Bosque*

Código de Actividad Límite	Día Límite de Entrega	Valor de Venta de la Unidades Entregadas	Ingreso al Momento de la Entrega
6	150	\$ 98.080.122,91	\$ 68.656.086,04
11	150	\$ 98.080.122,91	\$ 68.656.086,04
16	180	\$ 98.080.122,91	\$ 68.656.086,04
21	180	\$ 98.080.122,91	\$ 68.656.086,04
26	210	\$ 98.080.122,91	\$ 68.656.086,04
31	240	\$ 98.080.122,91	\$ 68.656.086,04
36	240	\$ 98.080.122,91	\$ 68.656.086,04
41	270	\$ 98.080.122,91	\$ 68.656.086,04
46	270	\$ 98.080.122,91	\$ 68.656.086,04
51	300	\$ 98.080.122,91	\$ 68.656.086,04

El día límite de entrega indica el día de finalización máximo de las actividades con las que se termina cada casa, para cumplir con el plazo de 6 meses. Por ejemplo la actividad 16 es “Acabados Casa 3”, con esta actividad se finaliza la construcción de la casa 3. Como esta unidad se vende en el mes 1, puede entregarse máximo el último día del período 6, es decir el día 180. Si la actividad 16 se terminara en el mes 5, el ingreso de \$ 68.656.086 se daría en el mes 5 y no en el mes 6 porque dicho valor se paga cuando la unidad es entregada al cliente.

Según el plan de ventas y el plazo de entregas establecido, se debe entregar un número mínimo de viviendas a partir del sexto mes. El siguiente cuadro resume la cantidad de casas a entregar en cada período:

Cuadro 11. Contiene la cantidad mínima de unidades a entregar por período para el proyecto *Portal del Bosque*

Periodo	Unidades Vendidas	Cantidad Mínima de Unidades a Entregar	Cantidad Mínima Acumulada de Unidades a Entregar
-1	2		
1	2		
2	1		
3	2		
4	2		
5	1	2	2
6		2	4
7		1	5
8		2	7
9		2	9
10		1	10

La cantidad mínima de casas a entregar en cada periodo corresponde a las unidades vendidas cuyo plazo de entrega se ha cumplido. La cantidad mínima acumulada de unidades a entregar se refiere a la cantidad total de casas que deben haber sido entregadas al final de cada período. Pero también debe tenerse en cuenta un número máximo de unidades a entregar por periodo, según se haya terminado de pagar la cuota inicial. Por ejemplo, la casa vendida en el período 2, su cuota inicial se termina de pagar en el sexto mes, pero el plazo de su entrega se cumple en el mes 7. Su entrega puede hacerse efectiva en el mes 6 si la unidad se ha terminado de construir; entonces ya no se entregarían 2 unidades, sino 3 ese mes.

Cuadro 12. Contiene la cantidad de cuotas iniciales que se terminan de pagar en cada periodo para el proyecto *Portal del Bosque*

Periodo	Unidades Vendidas	Cantidad Posible de Unidades a Entregar	Cantidad Máxima Acumulada de Unidades a Entregar
-1	2		
1	2		
2	1		
3	2		
4	2	2	2
5	1	2	4
6		1	5
7		2	7
8		2	9
9		1	10
10			

Cada casa se puede entregar a su propietario si éste ha terminado de pagar su cuota inicial, lo cual ocurre al quinto mes a partir de la venta, y lógicamente si la casa ya se ha terminado de construir. La cantidad posible de unidades a entregar se refiere a la cantidad de unidades cuya cuota inicial se terminan de pagar en cada período. Por ejemplo en el período 7 se culmina el pago de las dos casas vendidas en el tercer período. La cantidad máxima acumulada de unidades a entregar no es otra cosa que la cantidad total de viviendas cuyas cuotas iniciales ya se han pagado. Como la última casa se vende en el mes 5, la última cuota inicial se termina de pagar en el noveno mes; por eso en el período 9 pueden haberse entregado ya las 10 casas que componen el proyecto, así el compromiso sea entregar ésta última casa en el décimo período.

Según lo presupuestado, el *Portal del Bosque* financieramente se resume en las siguientes cifras:

Cuadro 13. Contiene el resumen del presupuesto para el proyecto *Portal del Bosque*

Costo Total	\$ 829.528.661
Valor de las Ventas	\$ 980.801.229
Utilidad e Imprevistos	\$ 151.272.568

La utilidad y los imprevistos se obtienen al restar el valor de las ventas menos el costo total. El valor de la utilidad y de los imprevistos está junto, debido a que en la etapa de planeación se espera que el costo del proyecto sea el presupuestado; cuando el costo presupuestado es igual al costo real, el valor destinado a los imprevistos pasa a ser parte de la utilidad. El valor del terreno está incluido dentro el costo total del proyecto.

Según el propósito del promotor del proyecto, puede hacer uso de la herramienta para los dos escenarios planteados desde el capítulo 2 de este documento: minimizar la inversión o endeudamiento (FCSP), o maximizar el valor presente neto de la inversión (MVPN). El proyecto de diez casas descrito en este capítulo, se empleó para estudiar las diferentes programaciones de obra que se obtienen con la herramienta computacional desarrollada en este trabajo de grado. El proyecto *Portal del Bosque* también se utilizó para la definición de las combinaciones entre operadores genéticos desarrollada en el capítulo 5.

6.2 ANÁLISIS DEL PROYECTO CON FCSP

6.2.1 Descripción de los parámetros del algoritmo genético empleados. El análisis sobre el uso de FCSP se hizo con varias ejecuciones de la herramienta, partiendo de diferentes poblaciones iniciales cada vez. Cada ejecución y su resultado (el resultado es una programación de obra) se nombró con un código:

Cuadro 14. Contiene la descripción de la población inicial en las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de FCSP

Código de Ejecución de la Herramienta	Descripción de la Población Inicial
MA1_FCSP	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores
R01_FCSP	La población inicial se construyó con valores aleatorios para la duración de las actividades y cero retraso entre actividades
MAR01_FCSP	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores y otros individuos con valores aleatorios para la duración de las actividades y cero retraso entre actividades
MAR02_FCSP	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores y otros individuos con valores aleatorios para la duración de las actividades y cero retraso entre actividades

Se emplearon los operadores genéticos determinados en el capítulo 5 para el uso de FCSP: Selección Proporcional y Elitismo, Recombinación 3 Padres 2 Puntos y Mutación Aleatoria, en cada una de las cuatro ejecuciones.

La diferencia entre una ejecución de la herramienta y otra no sólo radica en la población inicial, sino también en algunos parámetros de funcionamiento del algoritmo genético:

Cuadro 15. Contiene los parámetros del algoritmo genético empleados en las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de FCSP

Código de Ejecución de la Herramienta	Cantidad de Padres por Generación	Cantidad de Hijos por Generación	Cantidad de Hijos por Recombinación
MA1_FCSP	35	70	1
R01_FCSP	35	70	1
MAR01_FCSP	70	140	1
MAR02_FCSP	70	140	2

Otros parámetros que son iguales para las cuatro ejecuciones:

- Cantidad de Generaciones: 25
- Porcentaje de Mutación: 2%
- Porcentaje mínimo de Mutación: 1%
- Porcentaje de Elitismo: 80%

6.2.2 Análisis de resultados. Los egresos de dinero mes a mes, dependen del costo de las actividades que se están ejecutando; y los ingresos de dinero dependen de las ventas y de las entregas de unidades de vivienda. Cada resultado obtenido con la herramienta proporciona una programación de obra y el valor del flujo de caja acumulado mínimo:

Cuadro 16. Contiene el valor del flujo de caja acumulado mínimo obtenido con las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de FCSP

Código de Ejecución de la Herramienta	Flujo de Caja Acumulado Mínimo
MA1_FCSP	\$ -30.949.901
R01_FCSP	\$ -203.213.963
MAR01_FCSP	\$ -30.533.233
MAR02_FCSP	\$ -31.709.453

El mejor resultado es el obtenido con MAR01_FCSP. Esta solución exige un nivel de endeudamiento máximo de \$ 30.533.233 para ejecutar un proyecto cuyas utilidades se presupuestaron en un valor de \$ 151.272.568. La utilidad sobre la inversión sería casi del 400%:

$$Utilidad = \frac{\$151.272.568 - \$30.533.233}{\$30.533.233} \times 100\% = 395.44\%$$

El resultado es excelente, casi se cuadruplica el valor invertido en menos de un año.

Si bien las programaciones obtenidas son diferentes, muchas se parecen. El valor del flujo de caja acumulado mínimo para MA1_FCSP, MAR01_FCSP, MAR02_FCSP es muy parecido entre los tres, también lo es el ritmo de entrega de casas establecido mediante sus programaciones de obra:

Cuadro 17. Contiene el ritmo de entregas obtenido con las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de FCSP

Período	Código de Ejecución de la Herramienta	Cantidad Acumulada de Unidades Entregadas			
		MA1_FCSP	R01_FCSP	MARO1_FCSP	MARO2_FCSP
1					
2					
3					
4			4		
5		2	9	2	2
6		4	10	4	4
7		6		6	6
8		9		9	9
9		10		10	10
10					

Pareciera según los datos de este cuadro que las programaciones obtenidas con MA1_FCSP, MAR01_FCSP, MAR02_FCSP son idénticas, pero no es así. Este cuadro sólo muestra la cantidad total de unidades entregadas al final de cada periodo, lo cual es un

resultado de la programación de la obra y no la programación como tal. El valor del flujo de caja acumulado mínimo obtenido con cada una de estas ejecuciones así lo confirma, si bien son muy parecidos los valores, no son idénticos y esto se debe a que las tres programaciones son parecidas pero no iguales.

La solución R01_FCSP no sólo es la peor de todas sino que tiene implicaciones comerciales muy graves. Según esa programación de obra se deben terminar de entregar las diez casas en el mes 6 (Cuadro 17). Desde el punto de vista comercial es infactible pues la última casa se vende en el quinto mes, lo que quiere decir que a esa fecha sólo se habrán obtenido dos pagos de la cuota inicial de esa casa. Suponiendo que se pudiera hacer de esa forma, entregar las casas así no se hubiera dado el pago total de la cuota inicial. Que el pago del 70% se diera en el momento de la entrega así faltarían algunos pagos de la cuota inicial. Este escenario es el contemplado por la solución R01_FCSP. Sin embargo, de ser posible el supuesto, la solución sigue siendo mala. El valor del flujo de caja acumulado mínimo así lo evidencia. Los egresos de dinero superan a los ingresos de forma vertiginosa en los primeros meses, pues se construyen casi todas las casas al mismo tiempo sin ninguna necesidad de hacerlo. Por eso el flujo de caja acumulado mínimo de esta solución es tan elevado.

El resultado de la programación R01_FCSP es la entrega prematura de casas, lo cual es un fiasco desde el punto de vista financiero. En las dos últimas décadas del siglo XX fue muy usual ver proyectos que primero se construían y luego se vendían. Esta práctica ya no es tan común, y el motivo es el mismo que descalifica a la solución R01_FCSP. Todo el egreso se concentra en unos periodos, y el ingreso en otros. De esta forma se hace necesario contar con una gran capacidad de inversión o de endeudamiento para llevar a cabo el proyecto.

Con las otras tres programaciones se obtienen no solo excelentes resultados financieros sino programaciones perfectamente válidas. El ritmo de entregas obtenido, cumple a cabalidad con la cantidad mínima de unidades a entregar en cada período (Cuadro 11) y además cumple con la cantidad máxima de unidades a entregar planteada en el Cuadro 12.

6.3 ANÁLISIS DEL PROYECTO CON MVPN

6.3.1 Descripción de los parámetros del algoritmo genético empleados. Para este caso también se ejecutó la herramienta partiendo de cuatro poblaciones iniciales diferentes:

Cuadro 18. Contiene la descripción de la población inicial en las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de MVPN

Código de Ejecución de la Herramienta	Descripción de la Población Inicial
MA1_MVPN	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores
R01_MVPN	La población inicial se construyó con valores aleatorios para la duración de las actividades y cero retraso entre actividades
MAR01_MVPN	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores y otros individuos con valores aleatorios para la duración de las actividades y cero retraso entre actividades
MAR02_MVPN	La población inicial se construyó con los mejores individuos obtenidos de ejecuciones anteriores y otros individuos con valores aleatorios para la duración de las actividades y cero retraso entre actividades

Se emplearon los operadores genéticos determinados en el capítulo 5 para el uso de MVPN: Selección Torneo y Elitismo, Recombinación 2 Padres 1 Puntos y Mutación Aleatoria, en cada una de las cuatro ejecuciones.

Algunas diferencias entre las cuatro ejecuciones, en cuanto a los parámetros de funcionamiento del algoritmo genético se muestran en el cuadro 19:

Cuadro 19. Contiene los parámetros del algoritmo genético empleados en las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de MVPN

Código de Ejecución de la Herramienta	Cantidad de Padres por Generación	Cantidad de Hijos por Generación	Cantidad de Hijos por Recombinación
MA1_MVPN	30	60	1
R01_MVPN	30	60	1
MAR01_MVPN	64	128	1
MAR02_MVPN	64	128	2

El resto de parámetros son iguales para las cuatro ejecuciones:

- Cantidad de Generaciones: 25
- Porcentaje de Mutación: 15%
- Porcentaje mínimo de Mutación: 1%
- Porcentaje de Elitismo: 80%

6.3.2 Análisis de resultados. No siempre la programación que mejor valor presente neto ofrece sobre los flujos de caja netos de cada período es la mejor. Luego de ejecutar la herramienta se obtuvieron los resultados resumidos en el cuadro 20, con una tasa de descuento de 0.83% mensual.

Cuadro 20. Contiene el valor presente neto de los flujos de caja netos, obtenido con las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de MVPN

Código de Ejecución de la Herramienta	Valor Presente Neto de los flujos de caja netos
MA1_MVPN	\$ 103.672.768
R01_MVPN	\$ 140.257.066
MAR01_MVPN	\$ 103.672.769
MAR02_MVPN	\$ 103.672.765

Los resultados obtenidos con MAR01_MVPN, MAR02_MVPN y MA1_MVPN son exactamente iguales. Curiosamente se llegó al mismo resultado luego de ejecutar la

herramienta tres veces con diferentes poblaciones iniciales. Entonces se optó por hacer una ejecución más, partiendo de una población inicial generada aleatoriamente, utilizando los mismos parámetros que para MA1_MVPN. Se nombró esta ejecución con el código A1_MVPN. El valor presente neto obtenido fue: \$ 110.963.972. Para seguir con el análisis, es necesario observar el ritmo de entregas obtenido con las diferentes programaciones de obra expuesto en el cuadro 21.

Cuadro 21. Contiene el ritmo de entregas obtenido con las diferentes ejecuciones de la herramienta, para el proyecto *Portal del Bosque* con el uso de MVPN

Período	Código de Ejecución de la Herramienta	Cantidad de Unidades Entregadas por Período				
		MA1_MVPN	R01_MVPN	MARO1_MVPN	MARO2_MVPN	A1_MVPN
1						
2						
3						
4		3	10	3	3	5
5		1		1	1	1
6		3		3	3	1
7		1		1	1	1
8		2		2	2	2
9						
10						

En todas las programaciones resumidas en el cuadro 21, se entregan en el cuarto mes más unidades de las 2 que para ese período han terminado de pagar su cuota inicial (Cuadro 12). Como ya se dijo, desde el punto de vista comercial es imposible programar la obra así, porque no se puede entregar una casa a alguien que todavía no termina de pagarla. El impedimento de entregar la casa a un cliente que aún no termina de pagar la cuota inicial, es fundamentalmente el riesgo de que el cliente no termine de pagar todo el valor de la unidad. Al entregar la casa el propietario pasa a ser el cliente y deja de serlo el constructor; legalmente es muy difícil requerir el pago sobre un inmueble cuando la propiedad es de otra persona.

El motivo por el cual se obtienen resultados inviables, radica en la programación de la herramienta. Cuando se hace la evaluación de cada cromosoma, se obtienen los ingresos y egresos según la programación de obra codificada. El ingreso del 70% del valor de cada casa se asigna el día que se termina de construir la casa independientemente de que la cuota inicial se haya terminado de pagar o no. Como se busca maximizar el VPN de los flujos de caja netos, las mejores soluciones son aquellas con las que más rápido se obtienen importantes ingresos de dinero. Por eso el mejor VPN se obtuvo con la solución basada en individuos con retraso cero entre actividades.

Para solucionar esta situación se decidió hacer otra ejecución de la herramienta. En esta ejecución se modificó el código de la función de evaluación para garantizar que el ingreso del 70% del valor de cada casa, se diera sólo a partir del período en el que se termina de pagar la cuota inicial. A esta ejecución de la herramienta se le dio el código A2_MVPN. El valor presente neto obtenido fue \$ 102.401.136. El ritmo de entregas obtenido está en la Cuadro 22.

Cuadro 22. Contiene el ritmo de entregas obtenido con la ejecución A2_MVPN de la herramienta, para el proyecto *Portal del Bosque* con el uso de MVPN

Período	A2 MVPN	
	Cantidad de Unidades Entregadas por Período	Cantidad Acumulada de Unidades Entregadas
1		
2		
3		
4		
5	2	2
6	3	5
7	1	6
8	2	8
9	2	10
10		

La programación A2_MVPN cumple perfectamente con los compromisos de entregas (Cuadro 11) y con la restricción mediante la cual sólo se pueden entregar casas cuyas cuotas iniciales se hayan terminado de pagar (Cuadro 12). Para este caso sólo se pueden entregar las casas cuyas cuotas iniciales se hayan terminado de pagar. Entonces el terminar de construir una casa y tener que esperar para entregarla no es una buena opción. Por eso el resultado obtenido muestra que se debe entregar cada casa lo antes posible para obtener el ingreso total lo mas rápido posible, pero teniendo en cuenta que no se debe empezar a construir una casa si al terminarla es necesario esperar para poder entregarla.

6.4 COMPARACIÓN DE LOS RESULTADOS

Los dos mejores resultados (MAR01_FCSP y A2_MVPN) se evaluaron con ambas funciones objetivo. Al evaluar ambos resultados con la función objetivo FCSP, se obtuvo lo siguiente:

Cuadro 23. Contiene el valor del flujo de caja acumulado mínimo obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto *Portal del Bosque*

Código de Ejecución de la Herramienta	Flujo de Caja Acumulado Mínimo
MAR01_FCSP	\$ -30.533.233
A2_MVPN	\$ -30.473.710

Ambos resultados son muy parecidos, e igualmente buenos. Sucede lo mismo cuando se evalúan los dos resultados con la función objetivo MVPN:

Cuadro 24. Contiene el valor presente neto de los flujos de caja netos, obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto *Portal del Bosque*

Código de Ejecución de la Herramienta	Valor Presente Neto de los flujos de caja netos
MAR01_FCSP	\$ 102.398.801
A2_MVPN	\$ 102.401.134

Las dos programaciones de obra obtenidas no son iguales, eso se evidencia en el ritmo de entrega de una y otra (Cuadro 25), pero los valores del flujo de caja mínimo y el valor presente neto son muy parecidos entre sí.

Cuadro 25. Contiene el ritmo de entregas obtenido con las programaciones MA01_FCSP y A2_MVPN, del proyecto *Portal del Bosque*

Período	Código de Ejecución de la Herramienta	MAR01_FCSP		A2_MVPN	
		Cantidad de Unidades Entregadas	Cantidad Acumulada de Unidades Entregadas	Cantidad de Unidades Entregadas	Cantidad Acumulada de Unidades Entregadas
1					
2					
3					
4					
5		2	2	2	2
6		2	4	3	5
7		2	6	1	6
8		3	9	2	8
9		1	10	2	10
10					

Con ambas funciones objetivo se obtuvieron resultados muy parecidos, lo cual indica que para el proyecto *Portal del Bosque* se obtiene el mismo resultado independientemente de la función objetivo empleada. Para corroborarlo se hizo una ejecución adicional denominada MA2_MVPN. Esta ejecución se hizo con los mismos parámetros empleados en la ejecución MA1_MVPN (ver sección 6.3.1), pero se modificó la función de evaluación para restringir la entrega de casas sólo a partir del período en que se termina de pagar la cuota inicial. Con la programación MA2_MVPN se obtiene el mismo ritmo de entregas que con A2_MVPN, pero el valor presente neto obtenido es \$ 103.118.743. El flujo de caja acumulado mínimo obtenido con esta programación es \$ -30.473.710, igual que para A2_MVPN.

Los resultados obtenidos con MAR01_FCSP, MA2_MVPN y A2_MVPN son muy parecidos entre sí. El mejor resultado entre los tres, es el obtenido con MA2_MVPN; para el cual se empleo una población inicial basada en los mejores individuos obtenidos con ejecuciones previas. El segundo mejor resultado es el obtenido con A2_MVPN; para el cual se partió de una población inicial generada aleatoriamente. Finalmente, el resultado obtenido con MAR01_MVPN fue bueno pero no superó a los dos mencionados anteriormente; en esta ejecución se partió de una población inicial generada con los mejores individuos obtenidos en ejecuciones previas. Se obtuvieron mejores resultados con las ejecuciones en las que se empleó la función objetivo MVPN. En 25 generaciones se obtuvo un mejor resultado a partir de una población inicial generada aleatoriamente con A2_MVPN, que con MAR01_FCSP cuya población inicial contenía resultados de ejecuciones previas.

El uso de cualquiera de las dos funciones objetivo es indiferente en cuanto al resultado. La búsqueda del algoritmo genético tiene la misma dirección, no importa cual de las dos funciones objetivo se emplee. Pero se obtiene más rápidamente un buen resultado cuando se usa MVPN en vez de FCSP.

7. CONCLUSIONES Y RECOMENDACIONES

En general, y sobre el uso de los algoritmos genéticos se concluye que:

- Se demostró la aplicabilidad de los algoritmos genéticos como soporte en la toma de decisiones en la planeación de proyectos de construcción para el caso colombiano. Aún falta desarrollar su aplicabilidad en otros escenarios comunes de la construcción en el país, donde seguramente se tendrán consideraciones distintas y adicionales a las que se tuvieron para el desarrollo de este trabajo de grado.
- El campo de acción de los algoritmos genéticos debe abrirse aún más como soporte de toma de decisiones en el campo de la gestión de proyectos, no solamente en la solución de problemas de diseño en la ingeniería civil.
- El desarrollo de la herramienta computacional es prueba fehaciente de que los algoritmos genéticos son una herramienta muy valiosa y de gran aplicabilidad para la solución de diversos problemas en ingeniería civil. Con el estudio de caso desarrollado sobre el proyecto *Portal del Bosque* se demostró la aplicación directa de un algoritmo genético en la solución de un caso propio de la realidad colombiana. Los resultados obtenidos en este ejercicio, y consignados en este documento, sirven para validar la gran utilidad de los algoritmos genéticos, pero sobre todo la conveniencia de su uso en el caso colombiano.
- Se propuso el uso de un cromosoma con muchas ventajas, ya que ofrece soluciones siempre factibles y una gran probabilidad de transmisión del esquema.

Específicamente sobre el uso de la herramienta computacional se concluye que:

- Las dos funciones objetivo desarrolladas, encaminan la búsqueda del algoritmo genético en la misma dirección, pero es más eficaz el uso de MVPN que el uso de FCSP. Posiblemente se debe a que para FCSP se determinó el uso de recombinación entre tres padres con dos puntos de cruce y para MVPN se determinó entre dos padres con un solo punto de cruce. Como se observa en la sección 3.1.4, la probabilidad de transmisión del esquema es mayor cuando se utiliza recombinación entre dos padres con un punto de cruce.

- La función objetivo tiene mayor ingerencia sobre la eficacia de la búsqueda en el algoritmo genético que la población inicial. Así se observó en el estudio de caso desarrollado sobre el proyecto *Portal del Bosque*. Se obtuvo un mejor resultado utilizando la función objetivo MVPN a partir de una población inicial generada aleatoriamente, que empleando la función objetivo FCSP a partir de una población compuesta por individuos obtenidos con ejecuciones previas de la herramienta.

- Los resultados obtenidos al estudiar el caso del proyecto *Portal del Bosque* permiten concluir que si bien la herramienta proporciona un soporte muy valioso para la toma de decisiones en la etapa de planeación de proyectos, el criterio de quien hace uso de ella es indispensable para sacarle el mayor provecho.

8. ANOTACIONES SOBRE EL USO DE LA HERRAMIENTA BAJO EL AMBIENTE WINDOWS

El desarrollo de la herramienta computacional se hizo en base a Microsoft Excel debido a la estructura de datos que ofrece dicha aplicación. Posiblemente el desempeño del algoritmo genético sea mejor en otro ambiente diferente a Windows. Teóricamente nunca debería fallar el sistema al ejecutar la herramienta, pero existen evidencias sobre la inestabilidad del sistema operativo Windows cuando se exige demasiado al computador, o cuando se usa por extensos períodos de tiempo; la ejecución de un algoritmo genético tiene éstas implicaciones, aunque en el caso de éste trabajo de grado no se han evidenciado fallas en el sistema.

La preocupación por el tiempo de ejecución del algoritmo genético no debe darse al nivel de un fallo del sistema operativo, sino más bien sobre tiempo del que dispone el usuario para obtener una respuesta favorable haciendo uso de la herramienta computacional. Es muy difícil establecer bajo que parámetros de funcionamiento del algoritmo se hace inviable su funcionamiento en el ambiente Windows, es mejor establecer cual es la relación entre el tiempo de ejecución de la herramienta bajo este ambiente y el tiempo disponible por el usuario para obtener una respuesta. El tiempo de ejecución de la herramienta no depende solamente de los parámetros de funcionamiento del algoritmo genético, sino también de las características propias del proyecto tales como: cantidad de actividades, relaciones de precedencia entre actividades y composición de la estructura de costos. Adicionalmente influyen otros aspectos como las características de configuración del computador que se emplee. Por eso es imposible hacer una generalización sobre una posible inviabilidad del uso de la herramienta computacional en el ambiente Windows, dicha inviabilidad obedece a las necesidades y características propias de caso particular.

BIBLIOGRAFÍA

AKILESWARAN, V.; HAZEN, B. y MORIN, T.L. Complexity of the Project Sequencing Problem. Operations Research, Vol. 31, No 4. Julio – Agosto, 1983.

BOTTCHER, Jan; DREXL, Andreas; KOLISCH, Rainer y SALEWSKI, Frank. Project Scheduling under Partially Renewable Resource Constraints. Management Science, Vol. 45, No 4. Abril, 1999.

COELLO COELLO, Carlos A. Revista Red Científica [online]. [Citado: Septiembre 10 de 2004]. Disponible en <http://www.redcientifica.com/doc/doc199904260011.html>

COELLO COELLO, Carlos A.M; VAN VELDHUIZEN, David A. y LAMONT, Gary B. Evolutionary Algorithms for Solving Multi-Objective Problems. Plenum Publishers, 2000.

DOWSLAND, Kathryn A. Genetic Algorithms-A Tool for OR? [online]. [Citado Diciembre 1 de 2003]. The Journal of the Operational Research Society, Vol. 47 No. 4 (Apr., 1996). Disponible en: <http://links.jstor.org/sici?sici=0160-5682%28199604%2947%3A4%3C550%3AGATFO%3E2.0.CO%3B2-1>

DREXL, Andreas. Scheduling of Project Networks by Job Assingment. Management Science, Vol. 37, No 12. Diciembre, 1991.

EL-RAYES, Khaled and MOSELHI, Osama. Optimizing Resource Utilization for Repetitive Construction Projects. Journal of Construction Engineering and Management. ASCE. Enero/Febrero 2001

GOLDBERG E., David. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

GONÇALVES, José Fernando; MAGALHAES MENDES, Jorge José y RESENDE, Mauricio. A Genetic Algorithm for the Resource Constrained Mutli-Project Scheduling Problem. AT&T Labs Technical Report. Octubre 29, 2004

HARTMANN, Sönke. A Self-Adapting Genetic Algorithm for Project Scheduling under Resource Constraints. Naval Research Logistics 49:433-448, 2002.

_____. A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. Naval Research Logistics 45:733-750, 1988.

HEGAZY, Tarek and AYED, Amr. Simplified Spreadsheet Solutions: Models for Critical Path Method and Time-Cost-Tradeoff Analysis. Cost Engineering, Vol. 41/No. 7 Julio 1999.

HEGAZY, Tarek and ERSHAIN, Tolga. Simplified Spreadsheet Solutions. II: Overall Schedule Optimization. Journal of Construction Engineering and Management. ASCE. Noviembre/Diciembre 2001.

HEGAZY, Tarek and PETZOLD, Kevin. Genetic Optimization for Dynamic Project Control. Journal of Construction Engineering and Management. ASCE, Julio/Agosto, 2003.

HILLIER y LIEBERMAN. Investigación de Operaciones. Mc Graw Hill 2002

KALYANMOY, Deb. Multi-Objective Optimization using Evolutionary Algorithms. Willey, 2001.

LEE, Hyun-Soo and YI, Kyoo Jin. Application of Mathematical Matrix to Integrate Project Schedule and Cost. Journal of Construction Engineering and Management. ASCE. Septiembre/Octubre 1999.

LEU, Sou-Sen and YANG, Chung-Huei. GA-Based Multicriteria Optimal Model for Construction Scheduling. Journal of Construction Engineering and Management. ASCE. Noviembre/Diciembre, 1999.

LEU, Sou-Sen; YANG, Chung-Huei y HUANG, Jiun-Ching. Resource Leveling in Construction by Genetic Algorithm-based Optimization and its Decision Support System Application. Department of Construction Engineering, National Taiwan University of Science and Technology. Abril, 1999. Disponible en: <http://www.elsevier.com/locate/autocon>

LOO, S.M. WELLS, Earl. J.D. Algorithm Approach to Static Task Scheduling in a Reconfigurable Hardware Environment. Wingham. International Conference on Computers and Their Applications, Honolulu, Hawaii, Marzo, 26-28, 2003.

MANRIQUE RUIZ, Francisco. Señales del Futuro, Escenarios para La Industria de la Construcción. Presidencia CAMACOL, 2002

MOHRING, Rolf H. Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time. Operations Research, Vol. 32 No 1. Enero – Febrero, 1984

O'BRIEN, William J. and FISCHER, Martin A. Importance of Capacity Constraints to Construction Cost and Schedule. Journal of Construction Engineering and Management. ASCE. Septiembre/Octubre 2000.

RICH, Mark. Genetic Algorithms for Search. [Citado: Octubre 20 de 2004]. Disponible en: <http://www.cs.wisc.edu/~richm/cs540/notes/search-ga.html>

SANTO ORCERO, David. ¿Que es un Algoritmo Genético? [online]. [Citado: Septiembre 10 de 2004]. Disponible en: <http://www.orcero.org/irbis/disertacion/node189.html>. p. 3.

TAREK, Hegazy. Computer-Based Construction Project Management. Prentice Hall, 2003.

THESEN, Arne. Heuristic Scheduling of Activities under Resource and Precedence Restrictions. Management Science, Vol. 23, No 4. Diciembre, 1976.

TOKLU, Y Cengiz. Application of genetic Algorithms to Construction Scheduling with or without resource Constraints. Canadian Journal of Civil Engineering. Junio, 2002.

WHITLEY, Darrel. Genetic Algorithms and Evolutionary Computing [online]. [Citado Octubre 6 de 2004]. Computer Science Department, Colorado State University. Disponible a través de: whitley@cs.colostate.edu

Otras Referencias:

Frontline Systems and Spreadsheet Optimization, Frontline Systems, Inc Disponible en Internet: <http://www.solver.com/gabasics.htm#Selection>

Genetic Programming, Disponible en Internet: <http://www.genetic-programming.org/>

Profesor Kalyanmoy Deb, Home Page, Disponible en Internet: <http://www.iitk.ac.in/kangal/deb.htm>

Microsoft Developers Network, Disponible en Internet: <http://msdn.microsoft.com>

ANEXOS

ANEXO A

ESTUDIO DE CASO: EJEMPLO CONTROLADO

El desarrollo de un ejemplo controlado sirve como mecanismo para comprobar el funcionamiento del algoritmo genético y de toda la herramienta computacional. Cuando se trabaja con un proyecto pequeño, es decir, constituido por pocas actividades, se tiene un mayor conocimiento y dominio de la información. El desarrollo del ejemplo controlado consiste en hacer uso de la herramienta computacional con un proyecto pequeño. De esta forma se puede comparar el resultado obtenido mediante el uso del algoritmo genético y el resultado obtenido luego de realizar el ejercicio exhaustivamente para todo el espacio solución.

Características del proyecto:

Nombre: Controlado_1

El proyecto consiste en la realización de dos casas. Aunque el proyecto no existe en realidad, se creó con todas las consideraciones de un caso real.

Actividades que componen el proyecto:

Contiene las actividades que componen el proyecto *Controlado_1*.

Código de la Actividad	Descripción	Duración Máxima	Duración Mínima	Retraso Máximo
1	Preliminares	1	1	0
2	Casa 1	120	120	0
3	Casa 2	120	120	120

El precio de venta de cada unidad es: \$ 66.000.000 y las condiciones de pago son las siguientes:

- Cuota Inicial: \$ 33.000.000 (50% del precio de venta)
- Plazo de pago de la cuota inicial: Se debe cancelar el valor de la cuota inicial en el momento de la compra
- Pago al momento de la entrega de la unidad de vivienda: \$ 33.000.000 (30% del precio de venta)

Se propuso un plan de ventas según el cual ambas casas se venden en el mismo momento, pero el plazo de entrega es diferente para cada una de las unidades de vivienda. Al terminar la ejecución de la actividad 2 se termina la construcción de la Casa 1, el plazo máximo para terminarla es el día 121. Cuando se termina de ejecutar la actividad 3 se ha terminado de construir la Casa 2, el plazo máximo para terminar ésta actividad es el día 361.

Resumen del presupuesto:

Contiene el resumen del presupuesto para el proyecto *Controlado_1*

Costo Total	\$ 94.000.000
Valor de las Ventas	\$ 132.000.000
Utilidad e Imprevistos	\$ 38.000.000

El proyecto *Controlado_1* se sometió al uso de la herramienta computacional, y a un análisis exhaustivo de todas las soluciones posibles. Ambos análisis se hicieron empleando las dos funciones objetivo definidas, FCSP y MVPN.

Análisis Exhaustivo:

Según las restricciones de duración y retraso entre actividades del proyecto, todas las soluciones que comprenden el espacio solución pueden representarse en 121 cromosomas. La duración de las tres actividades es fija; el retraso de la actividad 1 y el retraso de la actividad 2, es de cero días. Solamente el retraso de la actividad 3 tiene un rango de [0, 120] días. Los cromosomas que pueden representar una programación de este proyecto son los siguientes:

Cromosomas que componen el espacio solución del proyecto *Controlado_1*

		Cromosoma 1	Cromosoma 2	Cromosoma 3	...	Cromosoma 121
Actividad 1	Duración	1	1	1	1	1
	Retraso	0	0	0	0	0
Actividad 2	Duración	120	120	120	120	120
	Retraso	0	0	0	0	0
Actividad 3	Duración	120	120	120	120	120
	Retraso	0	1	2	...	120

Luego del análisis exhaustivo con ambas funciones objetivo se encontró el mismo óptimo. La programación óptima esta representada por el siguiente cromosoma:

Cromosoma que representa la solución óptima del proyecto *Controlado_1*

		Cromosoma 120
Actividad 1	Duración	1
	Retraso	0
Actividad 2	Duración	120
	Retraso	0
Actividad 3	Duración	120
	Retraso	119

Con este cromosoma se obtuvieron los siguientes resultados:

Contiene el flujo de caja mínimo acumulado y el valor presente neto de los flujos de caja netos, obtenidos con la solución óptima del proyecto *Controlado_1*

Flujo de Caja Acumulado Mínimo	\$ 15.999.999,00
Valor Presente Neto de los Flujos de Caja Netos	\$ 33.732.088,97

La tasa de descuento empleada es de 6% mensual. Es una tasa muy alta, pero se escogió así para observar fácilmente la diferencia entre los resultados buenos y los malos.

Determinación de los mejores operadores genéticos:

La determinación de los mejores operadores genéticos se realizó de forma independiente con cada una de las funciones objetivo. Se definió un esquema de decisión para la comparación entre los operadores genéticos:

Esquema de decisión para la comparación entre operadores genéticos

Comparación entre Métodos de Selección:

Torneo:	1. Torneo - Mejores padres e hijos	Método de Selección Escogido
	2. Torneo - Elitismo	
Proporcional:	3. Proporcional - Mejores padres e hijos	
	4. Proporcional - Elitismo	
Escalada:	5. Escalada - Mejores padres e hijos	
	6. Escalada - Elitismo	

Comparación entre Operadores de Recombinación:

<i>Método de Selección Escogido</i>	7. 2 Padres 1 Punto	Operador de Recombinación Escogido
	8. 2 Padres 2 Puntos	
	9. 3 Padres 2 Puntos	

Comparación entre Operadores de Mutación:

<i>Operador de Recombinación Escogido</i>	10. Aleatoria	Operador de Mutación Escogido
	11. Incremento - Decremento	
	12. Variable	

Mediante el orden establecido en este esquema, se compran los operadores genéticos bajo las mismas condiciones y luego se decide cual es el mejor. Durante la comparación entre los métodos de selección se utilizan siempre los operadores recombinação 2 Padres 1 Punto y Mutación Aleatoria. La comparación entre operadores de recombinação se hace siempre con Mutación Aleatoria.

Los siguientes parámetros se dejaron fijos para todas las comparaciones:

Tamaño de la población = 10

Cantidad de generaciones = 5

Cantidad de hijos por generación = 10

Cantidad de hijos por recombinação = 1

Porcentaje de Mutación = 2% para FCSP y 15% para MVPN

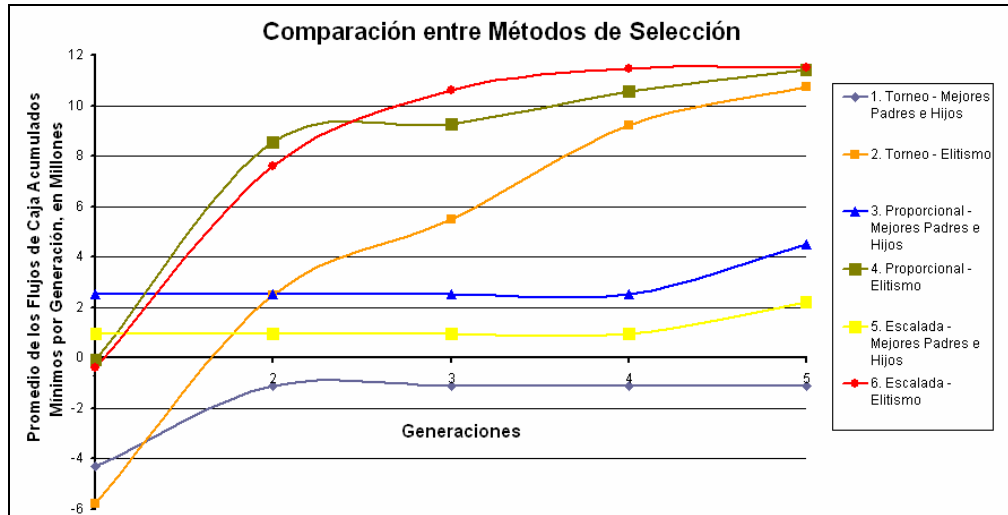
Porcentaje mínimo de mutación = 1%

Elitismo = 15%; éste porcentaje se escogió bajo debido al tamaño tan reducido de la población y a la poca cantidad de generaciones. Si el porcentaje de elitismo llegase a ser muy alto podrían ignorarse hijos muy buenos.

Determinación de los operadores genéticos con el uso de FCSP:

Al comparar los diferentes métodos de selección, se evidencia el gran desempeño que se logra del algoritmo genético cuando se utiliza el elitismo.

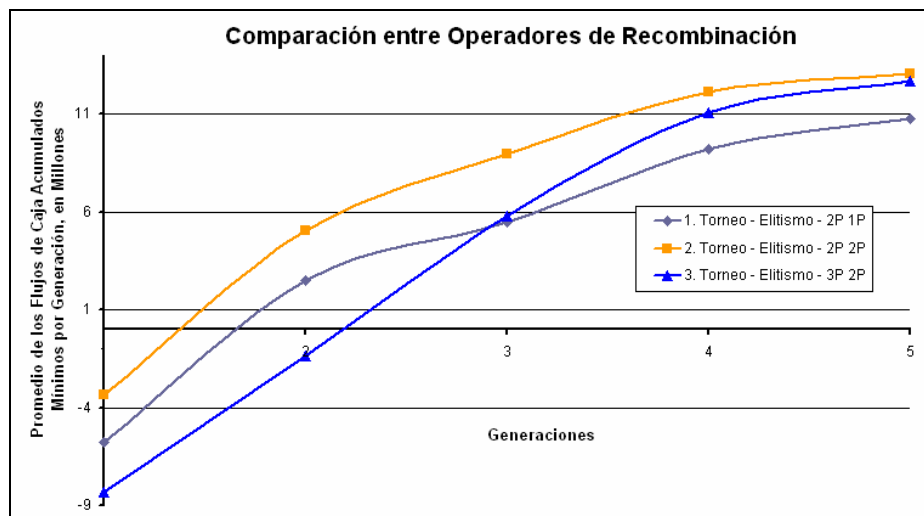
Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de FCSP



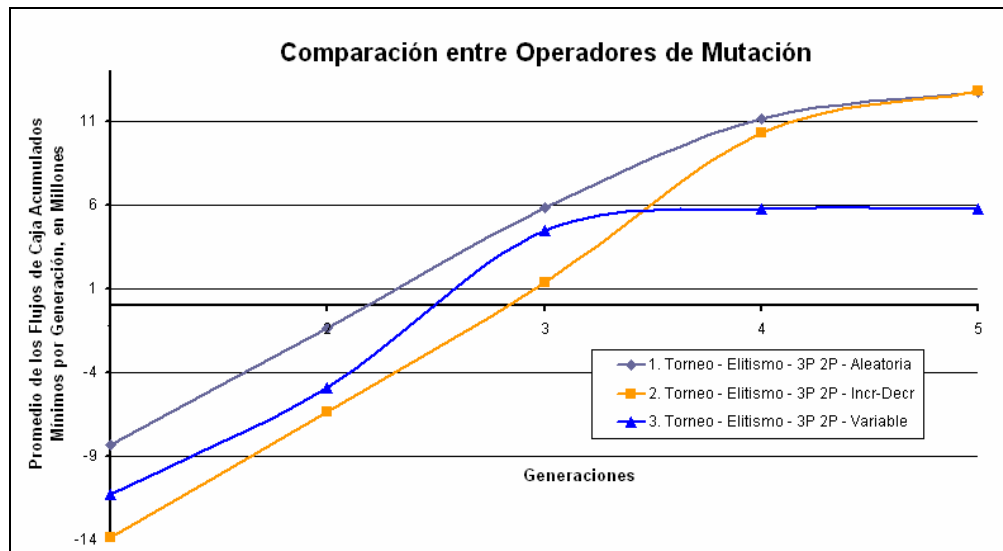
Los mejores resultados se obtienen con las combinaciones: Torneo – Elitismo, Proporcional – Elitismo y Escalada – Elitismo. Sin embargo el mejor desempeño del algoritmo se obtuvo con la combinación Torneo – Elitismo pues se logro en la quinta generación un valor muy similar al de las otras dos combinaciones, aún cuando la población inicial era peor.

El operador de recombinación escogido es 3 padres 2 puntos de cruce porque el mejor desempeño del algoritmo se logra con este operador.

Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de FCSP



Proceso evolutivo obtenido con los diferentes operadores de mutación para el uso de FCSP

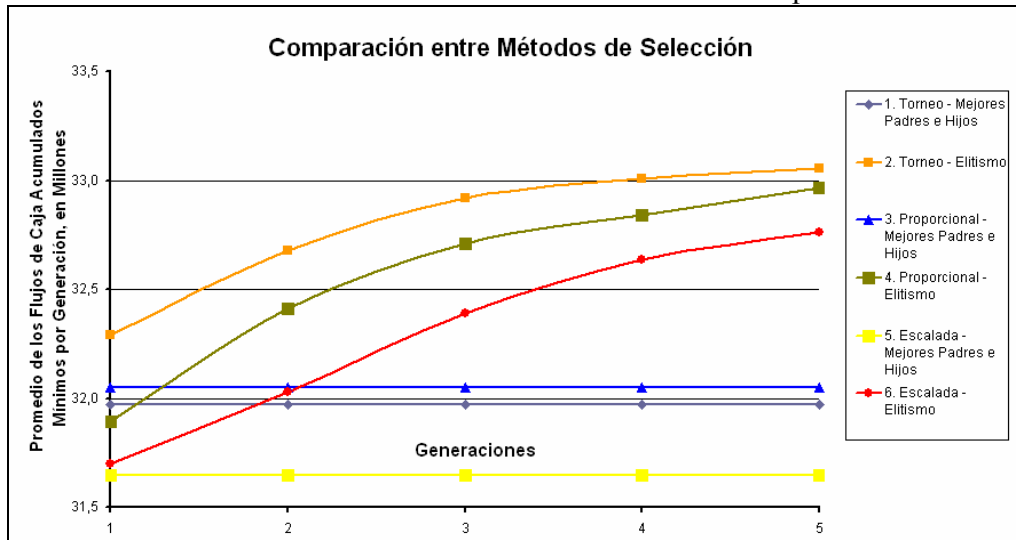


Como se evidencia en la gráfica anterior, el mejor operador de mutación es el Incremento-Decremento.

Los operadores escogidos para el uso de FCSP son:
Método de Selección: Torneo – Elitismo
Operador de Recombinación: 3 Padres 2 Puntos de Cruce
Operador de Mutación: Incremento-Decremento

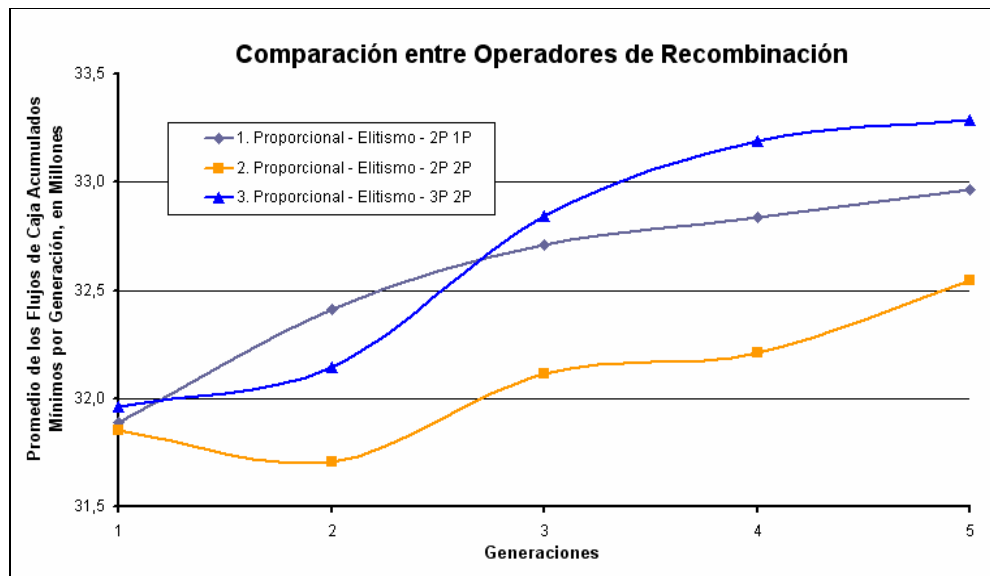
Determinación de los operadores genéticos con el uso de MVPN:
La combinación entre Torneo y Elitismo es muy buena, sin embargo la combinación Proporcional – Elitismo es mejor.

Proceso evolutivo obtenido con los diferentes métodos de selección para el uso de MVPN



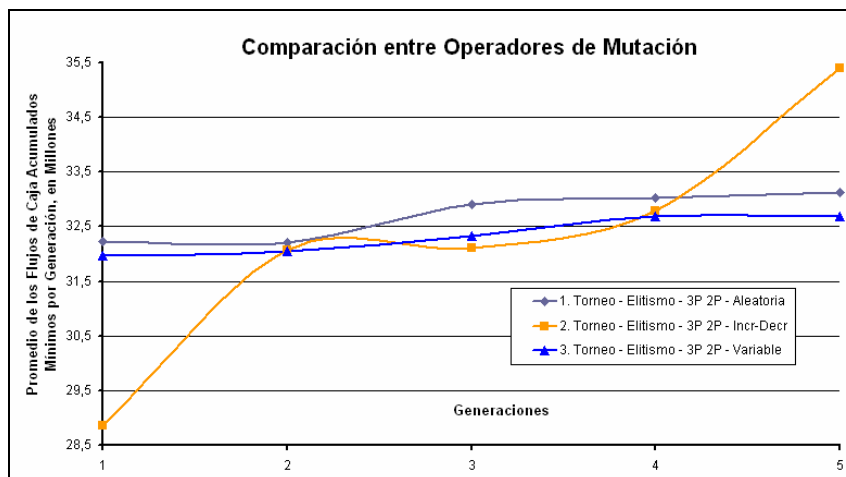
Partiendo de una población inicial más mala que la de Torneo – Elitismo, la combinación Proporcional – Elitismo obtuvo un resultado muy parecido. Por eso el método de selección escogido es Proporcional – Elitismo.

Proceso evolutivo obtenido con los diferentes operadores de recombinación para el uso de MVPN



El mejor proceso evolutivo se obtuvo con el operador de recombinación 3 padres 2 puntos de cruce.

Proceso evolutivo obtenido con los diferentes operadores de mutación para el uso de MVPN



La mutación Incremento – Decremento supero notablemente el desempeño de las otras dos mutaciones.

Los operadores escogidos para el uso de FCSP son:

Método de Selección: Proporcional – Elitismo

Operador de Recombinación: 3 Padres 2 Puntos de Cruce

Operador de Mutación: Incremento-Decremento

Análisis con el Algoritmo Genético:

La búsqueda de la solución óptima de forma exhaustiva requirió el análisis de 121 soluciones posibles, con cada una de las funciones objetivo. Con el algoritmo genético se logra encontrar ese optimo haciendo una cantidad menor de cálculos.

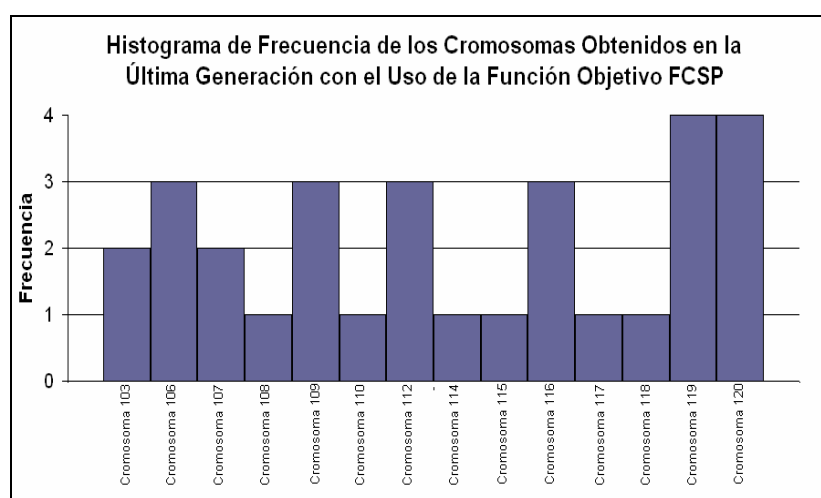
Se realizaron 30 ejecuciones de la herramienta utilizando la Función Objetivo FCSP y los operadores escogidos en la fase anterior. Los resultados son los siguientes:

Contiene los resultados obtenidos en la última generación con el uso de la función objetivo FCSP en el proyecto *Controlado 1*

Cromosoma	Flujo de Caja Acumulado Mínimo	Frecuencia
Cromosoma 103	\$ 9.766.666	2
Cromosoma 106	\$ 10.866.666	3
Cromosoma 107	\$ 11.233.332	2
Cromosoma 108	\$ 11.599.999	1
Cromosoma 109	\$ 11.966.665	3
Cromosoma 110	\$ 12.333.333	1
Cromosoma 112	\$ 13.066.665	3

Cromosoma 114	\$ 13.799.999	1
Cromosoma 115	\$ 14.166.665	1
Cromosoma 116	\$ 14.533.333	3
Cromosoma 117	\$ 14.899.999	1
Cromosoma 118	\$ 15.266.665	1
Cromosoma 119	\$ 15.633.333	4
Cromosoma 120	\$ 15.999.999	4

Histograma de frecuencia de los cromosomas obtenidos en la última generación con el uso de la función objetivo FCSP en el proyecto *Controlado_1*



El cromosoma 120, el óptimo, fue el resultado del 13.3% de las 30 ejecuciones de la herramienta. Adicionalmente fue el resultado más frecuente que se obtuvo en la última generación de estas 30 ejecuciones.

Con la función objetivo MVPN se realizaron también 30 ejecuciones. Siempre se obtuvieron resultados sobresalientes:

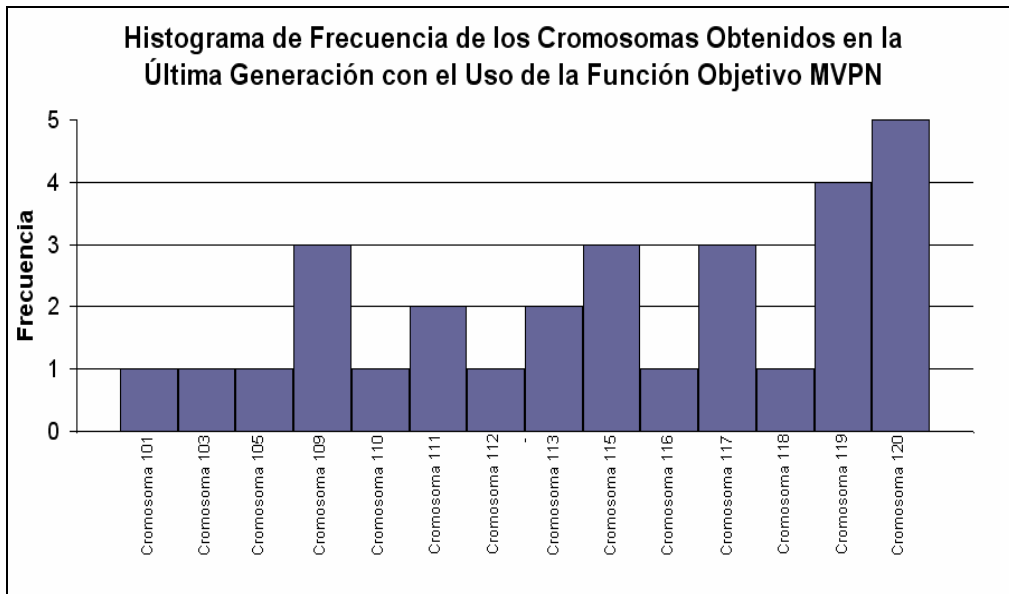
Contiene los resultados obtenidos en la última generación con el uso de la función objetivo MVPN en el proyecto *Controlado_1*

Cromosoma	Valor Presente Neto de los Flujos de Caja Netos	Frecuencia
Cromosoma 101	\$ 32.584.811	1
Cromosoma 103	\$ 32.705.578	1
Cromosoma 105	\$ 32.826.343	1
Cromosoma 109	\$ 33.067.876	3
Cromosoma 110	\$ 33.128.259	1
Cromosoma 111	\$ 33.188.642	2
Cromosoma 112	\$ 33.249.025	1

Cromosoma 113	\$ 33.309.409	2
Cromosoma 115	\$ 33.430.174	3
Cromosoma 116	\$ 33.490.557	1
Cromosoma 117	\$ 33.550.939	3
Cromosoma 118	\$ 33.611.323	1
Cromosoma 119	\$ 33.671.708	4
Cromosoma 120	\$ 33.732.089	5

Fuera de que siempre se obtuvieron buenos resultados, el más frecuente fue el óptimo (Cromosoma 120).

Histograma de frecuencia de los cromosomas obtenidos en la última generación con el uso de la función objetivo MVPN en el proyecto *Controlado_1*



El resultado óptimo obtenido a través de ambas funciones objetivo es el mismo, sin embargo parece ser más provechoso el uso de MVPN pues se encontró más veces el cromosoma 120 con esta función objetivo.

Encontrar el óptimo con el algoritmo genético sólo implicó analizar 100 cromosomas, mientras que exhaustivamente fue necesario analizar todos los 121 que componen el espacio solución. Son 100 cromosomas pues el algoritmo genético se ejecutó para 5 generaciones, cada una con 10 padres y 10 hijos $((10 + 10) \times 5 = 100)$.

ANEXO B

CÓDIGO DE PROGRAMACIÓN DE LA HERRAMIENTA COMPUTACIONAL

El siguiente código esta escrito en lenguaje Visual Basic. Contiene todas las funciones que componen la herramienta.

Dim min As Single

.....

'Genera una Poblacion inicial:

'Selecciona los Nind mejores cromosomas de la Hoja Resultados_Previos

'y los copia en la Hoja Poblacion

.....

Public Sub PobIni_ResHistoric1()

Dim Tpob, Nact, fila, Nind, aux, j, Tpobv As Integer

Dim Xp As Single

Dim Elite() As Integer 'Vector en el que se guardan las posiciones
'de los Nind individuos seleccionados

Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion

Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades

Ngenes = Nact * 2 'Número de genes

Nind = Tpob 'Número de Individuos a copiar

'Cuenta cuantos individuos hay en la hoja Resultados_Previos

j = 1

While (Not (IsEmpty(Worksheets("Resultados_Previos").Cells(1, j))))

 Tpobv = j

 j = j + 1

Wend

ReDim Elite(Tpobv) 'Cambio de tamaño según Tamaño Población

'Inicialización vector Elite

For i = 1 To Tpobv

 Elite(i) = i

Next i

'Ordenar para escoger los mejores Nind individuos de la población

For i = 1 To Nind

 For j = i + 1 To Tpobv

 If (Worksheets("Resultados_Previos").Cells(1, Elite(i)) < Worksheets("Resultados_Previos").Cells(1,
Elite(j))) Then

 aux = Elite(i): Elite(i) = Elite(j): Elite(j) = aux

 End If

 Next j

Next i

Worksheets("Población").Range("A1:AZ10000").ClearContents

```

'Copiar los Nind individuos a la hoja Pob_Intermedia
For i = 1 To Nind 'Recorre el vector Elite

    For fila = 1 To (Ngenes + 2) Step 2 'Recorre la hoja Pob_Intermedia
        'Copia el individuo en la posicion Elite (i)
        'de la Hoja Poblacion a la Hoja Pob_Intermedia
        Worksheets("Población").Cells(fila, i).Value = Worksheets("Resultados_Previos").Cells(fila,
Elite(i)).Value
        Worksheets("Población").Cells(fila + 1, i).Value = Worksheets("Resultados_Previos").Cells(fila + 1,
Elite(i)).Value
        Next fila
    Next i
End Sub

.....

'Genera una Poblacion Inicial con Numeros Aleatorios en la Hoja Población
.....
Public Sub PobIni_Aleatorio()

    Dim Tpob, Nact, fila, col, act, Li, Ls, D, R, RM As Integer

    Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
    Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades
    Ngenes = Nact * 2 'Número de genes

    Randomize

    Worksheets("Población").Range("A1:AZ10000").ClearContents

    'Crea la Poblacion Inicial en la Hoja Población
    For col = 1 To Tpob 'Recorre la Hoja Actividades

        act = 2

        For fila = 3 To (Ngenes + 2) Step 2 'Recorre la Hoja Población

            'Obtiene la Dur max y min de las actividades y retraso maximo
            Li = Worksheets("Actividades").Cells(act, 6).Value 'duracion minima
            Ls = Worksheets("Actividades").Cells(act, 5).Value 'duracion maxima
            RM = Worksheets("Actividades").Cells(act, 7).Value 'retraso maximo
            act = act + 1
            'Asigna la duración y retraso para cada actividad
            D = Int((Ls - Li + 1) * Rnd + Li)
            Worksheets("Población").Cells(fila, col).Value = D
            R = Int(Rnd * RM) 'ojo: aqui se modifica el rango de las holguras
            Worksheets("Población").Cells(fila + 1, col).Value = R

        Next fila
    Next col
End Sub

.....

'Genera una Población Inicial con Dur aleatoria y retraso = 0 en la Hoja Población
.....
Public Sub PobIni_Retraso0()

```

```

Dim Tpob, Nact, fila, col, act, Li, Ls, D As Integer

Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades
Ngenes = Nact * 2 'Número de genes

Randomize

Worksheets("Población").Range("A1:AZ10000").ClearContents

'Crea la Poblacion Inicial en la Hoja Población
For col = 1 To Tpob 'Recorre la Hoja Actividades

    act = 2

    For fila = 3 To (Ngenes + 2) Step 2 'Recorre la Hoja Población

        'Obtiene la Dur max y min de las actividades
        Li = Worksheets("Actividades").Cells(act, 6).Value
        Ls = Worksheets("Actividades").Cells(act, 5).Value
        act = act + 1
        'Asigna la duración y retraso para cada actividad
        D = Int((Ls - Li + 1) * Rnd + Li)
        Worksheets("Población").Cells(fila, col).Value = D
        Worksheets("Población").Cells(fila + 1, col).Value = 0

    Next fila

Next col

End Sub

'-----
'Selecciona por torneos aleatorios los cromosomas de la Hoja Población y
'copia en la Hoja Pob_Intermedia la posicion de los seleccionados
'-----
Public Sub Selección_Torneo()

    Dim Tpob, Nact, fila, i, Ia, Ib, Ngenes, Nhijos, Nhr, Ppr, CS, Pos() As Integer
    Dim Aa, Ab As Single

    Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
    Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades
    Ngenes = Nact * 2 'Número de genes
    Nhijos = Worksheets("Control").Cells(7, 2).Value 'cantidad de hijos por generación
    Nhr = Worksheets("Control").Cells(8, 2).Value 'cantidad de hijos por recombinacion
    Ppr = Worksheets("Control").Cells(16, 2).Value 'cantidad de padres por recombinación
    CS = (Nhijos / Nhr + 1) * Ppr 'cantidad de padres a seleccionar
    ReDim Pos(CS)

    Randomize

' hace el primer torneo para seleccionar el primer individuo
' Produce posociones aleatorias entre las cuales se hace el torneo

```

```

Ia = Int(Tpob * Rnd + 1)
Ib = Int(Tpob * Rnd + 1)
If Ia = Ib Then
    Ia = Int((Tpob / 2) + (Tpob / 2) * Rnd)
    Ib = Int((Tpob / 2) * Rnd + 1)
End If
'Obtiene los valores de aptitud de los cromosomas en las posiciones Ia Ib
Aa = Worksheets("Población").Cells(1, Ia).Value
Ab = Worksheets("Población").Cells(1, Ib).Value

'Copia en la Hoja Pob_Intermedia al ganador del torneo
If Aa > Ab Then
    Pos(1) = Ia
End If

If Ab > Aa Then
    Pos(1) = Ib
End If

If Aa = Ab Then
    Pos(1) = Ib
End If

'hace torneos entre dos padres, y guarda la posicion del ganador en pos
For i = 2 To CS
    'Produce posociones aleatorias entre las cuales se hace el torneo
    Ia = Int(Tpob * Rnd + 1)
    Ib = Int(Tpob * Rnd + 1)

    If Ia = Ib Then
        Ia = Int((Tpob / 2) + (Tpob / 2) * Rnd)
        Ib = Int((Tpob / 2) * Rnd + 1)
    End If
    'Obtiene los valores de aptitud de los cromosomas en las posiciones Ia Ib
    Aa = Worksheets("Población").Cells(1, Ia).Value
    Ab = Worksheets("Población").Cells(1, Ib).Value

    'Copia en la Hoja Pob_Intermedia al ganador del torneo
    If Aa > Ab Then
        If Ia <> Pos(i - 1) And Ia <> Pos(i - 2) Then
            Pos(i) = Ia
        Else
            i = i - 1
        End If
    End If

    If Ab > Aa Then
        If Ib <> Pos(i - 1) And Ib <> Pos(i - 2) Then
            Pos(i) = Ib
        Else
            i = i - 1
        End If
    End If
End For

```

```

If Aa = Ab Then
  If Ib <> Pos(i - 1) And Ib <> Pos(i - 2) Then
    Pos(i) = Ib
  Else
    i = i - 1
  End If
End If
Next i

Worksheets("Pob_Intermedia").Range("A1:IV10000").ClearContents

For i = 1 To CS
  Worksheets("Pob_Intermedia").Cells(i, 1).Value = Pos(i)
Next i

End Sub

'=====
'Selecciona los mejores cromosomas proporcionalmente a su valor de aptitud
'y copia la posicion de los seleccionados en Pob_Intermedia
'"Ruleta"
'=====
Public Sub Selección_Proporcional()

  Dim Tpob, Nact, fila, Nind, i, poss, Ngenes, Nhijos, Nhr, Ppr, CS As Integer
  Dim Vect() As Single 'Vector en el que se guardan los rangos
    'de aptitud de los individuos
  Dim Pos() As Integer 'Guarda las posiciones de los individuos seleccionados
  Dim Sapt, Rand As Single

  Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
  Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades
  Ngenes = Nact * 2 'Número de genes
  Nhijos = Worksheets("Control").Cells(7, 2).Value 'cantidad de hijos por generación
  Nhr = Worksheets("Control").Cells(8, 2).Value 'cantidad de hijos por recombinacion
  Ppr = Worksheets("Control").Cells(16, 2).Value 'cantidad de padres por recombinación
  CS = (Nhijos / Nhr + 1) * Ppr 'cantidad de padres a seleccionar

  ReDim Vect(Tpob) 'Cambio de tamaño según Tamaño Población
  ReDim Pos(CS) 'Cambio de tamaño según la cantidad de padres a seleccionar

  'Suma las aptitudes de todos los cromosomas en Sapt y
  'guarda las aptitudes de todos los cromosomas en Vect
  Sapt = 0
  For i = 1 To Tpob
    Sapt = Sapt + Worksheets("Población").Cells(1, i).Value 'Guarda la sumatoria de las aptitudes de todos
los cromosomas
    Vect(i) = Worksheets("Población").Cells(1, i)
  Next i

  'Asigna el limte superior del rango de cada individuo y lo guarda en Vect
  Vect(1) = Vect(1) / Sapt
  For i = 2 To Tpob

```

```

    Vect(i) = Vect(i) / Sapt
    Vect(i) = Vect(i - 1) + Vect(i)
Next i

Randomize

'selecciona el primer padre
Rand = Rnd 'Genera un # aleatorio
j = 0
While (j < Tpob)
    If Vect(j + 1) >= Rand And Vect(j) < Rand Then 'Pregunta si el aleatorio esta en el
                                                'intervalo del individuo en la posicion (j+1)
        Pos(1) = (j + 1) 'Asigna a Pos la posicion del individuo seleccionado
        j = Tpob
    Else
        j = j + 1
    End If
Wend

'Selecciona los individuos y guarda la posicion de los seleccionados en Pos
For i = 2 To CS
    Rand = Rnd 'Genera un # aleatorio
    j = 0
    While (j < Tpob)
        If Vect(j + 1) >= Rand And Vect(j) < Rand Then 'Pregunta si el aleatorio esta en el
                                                    'intervalo del individuo en la posicion (j+1)
            poss = (j + 1)
            If poss <> Pos(i - 1) And poss <> Pos(i - 2) Then 'si el individuo seleccionado es diferente al
seleccionado anteriormente, entonces lo selecciona
                Pos(i) = (j + 1) 'Asigna a Pos la posicion del individuo seleccionado
                j = Tpob
            Else
                i = i - 1
                j = Tpob
            End If
        Else
            j = j + 1
        End If
    Wend
Next i

Worksheets("Pob_Intermedia").Range("A1:IV10000").ClearContents

'Copiar la posicion de los individuos seleccionados en Pob_Intermedia
For i = 1 To CS 'Recorre el vector Pos
    Worksheets("Pob_Intermedia").Cells(i, 1).Value = Pos(i)
Next i

End Sub

.....
'Seleccion Escalonada:
'Organiza los individuos, traza una recta entre el ind con menor aptitud y el ind
'con mayor aptitud; luego prolonga la aptitud de cada individuo hasta la recta

```

```
Public Sub Selección_Escalonada()
```

```
Dim Tpob, Nact, fila, Nind, i, poss, Ngenes, Nhijos, Nhr, Ppr, CS As Integer  
Dim Vect() As Single 'Vector en el que se guardan los rangos  
                  'de aptitud de los individuos  
Dim Pos() As Integer 'Guarda las posiciones de los individuos seleccionados  
Dim Sapt, Rand As Single
```

```
Dim Elite() As Integer 'Vector en el que se guardan las posiciones  
                  'de los Nind individuos seleccionados
```

```
Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion  
Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades  
Ngenes = Nact * 2 'Número de genes  
Nhijos = Worksheets("Control").Cells(7, 2).Value 'cantidad de hijos por generación  
Nhr = Worksheets("Control").Cells(8, 2).Value 'cantidad de hijos por recombinacion  
Ppr = Worksheets("Control").Cells(16, 2).Value 'cantidad de padres por recombinación  
CS = (Nhijos / Nhr + 1) * Ppr 'cantidad de padres a seleccionar
```

```
ReDim Vect(Tpob) 'Cambio de tamaño según Tamaño Población  
ReDim Pos(CS) 'Cambio de tamaño según cantidad de padres a seleccionar  
ReDim Elite(Tpob) 'Cambio de tamaño según Tamaño Población
```

```
'Inicialización vector Elite  
For i = 1 To Tpob  
    Elite(i) = i  
Next i
```

```
'Ordena los individuos de la población de mejor a peor en el vector Elite  
'OJO: ordena las posiciones, no los individuos físicamente  
For i = 1 To Tpob  
    For j = i + 1 To Tpob  
        If (Worksheets("Población").Cells(1, Elite(i)) < Worksheets("Población").Cells(1, Elite(j))) Then  
            aux = Elite(i): Elite(i) = Elite(j): Elite(j) = aux  
        End If  
    Next j  
Next i
```

```
'guarda las aptitudes de todos los cromosomas en Vect  
For i = 1 To Tpob  
    Vect(i) = Worksheets("Población").Cells(1, Elite(i))  
Next i
```

```
Amen = Vect(Tpob) 'Menor aptitud  
Amax = Vect(1) 'Mayor aptitud
```

```
'Prolonga la aptitud de cada individuo hasta la recta  
'y guarda dichos valores en Vect  
'Suma las nuevas aptitudes en Sapt  
Sapt = 0
```

```
For i = 0 To (Tpob - 1)  
    Vect(i + 1) = (((Amax - Amen) / Tpob) * (Tpob - i))  
    Sapt = Sapt + Vect(i + 1) 'Guarda la sumatoria de las aptitudes de todos los cromosomas
```



```

Next i

'Asigna el limite superior del rango de cada individuo y lo guarda en Vect
Vect(1) = Vect(1) / Sapt
For i = 2 To Tprob
    Vect(i) = Vect(i) / Sapt
    Vect(i) = Vect(i - 1) + Vect(i)
Next i

Randomize
'Selecciona el primer padre
Rand = Rnd 'Genera un # aleatorio
j = 0
While (j < Tprob)
    If Vect(j + 1) >= Rand And Vect(j) < Rand Then 'Pregunta si el aleatorio esta en el
                                                'intervalo del individuo en la posicion (j+1)
        Pos(1) = Elite(j + 1)
        j = Tprob
    Else
        j = j + 1
    End If
Wend

'Selecciona los individuos y guarda la posicion de los seleccionados en Pos
For i = 2 To CS
    Rand = Rnd 'Genera un # aleatorio
    j = 0
    While (j < Tprob)
        If Vect(j + 1) >= Rand And Vect(j) < Rand Then 'Pregunta si el aleatorio esta en el
                                                'intervalo del individuo en la posicion (j+1)
            poss = Elite(j + 1)
            If Pos(i - 1) <> poss And Pos(i - 2) <> poss Then 'si el individuo seleccionado es diferente a los dos
seleccionados anteriormente, entonces lo selecciona
                Pos(i) = Elite(j + 1) 'Asigna a Pos la posicion del individuo seleccionado
                j = Tprob
            Else
                i = i - 1
                j = Tprob
            End If
        Else
            j = j + 1
        End If
    Wend
Next i

Worksheets("Pob_Intermedia").Range("A1:IV10000").ClearContents

'Copiar la posicion de los individuos seleccionados a Pob_Intermedia
For i = 1 To CS 'Recorre el vector Pos
    Worksheets("Pob_Intermedia").Cells(i, 1).Value = Pos(i)
Next i

End Sub

```

```

'.....
'Selecciona individuos aleatoriamente de la hoja Población y copia la posicion
'de los seleccionados en la hoja Pob_Intermedia
'.....
Public Sub Selección_Aleatoria()

    Dim Tpob, Pos, CS, Nhijos, Nhr, Ppr As Integer

    Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
    Nhijos = Worksheets("Control").Cells(7, 2).Value 'cantidad de hijos por generación
    Nhr = Worksheets("Control").Cells(8, 2).Value 'cantidad de hijos por recombinacion
    Ppr = Worksheets("Control").Cells(16, 2).Value 'cantidad de padres por recombinación
    CS = (Nhijos / Nhr + 1) * Ppr 'cantidad de padres a seleccionar

    Randomize
    For i = 1 To CS
        If i = 1 Then 'cuando selecciona el primero
            Worksheets("Pob_Intermedia").Cells(i, 1).Value = Int((Tpob - 1) * Rnd + 1)
        End If
        If i = 2 Then 'cuando selecciona el segundo
            Pos = Int((Tpob - 1) * Rnd + 1)
            If Pos <> Worksheets("Pob_Intermedia").Cells(i, 1).Value Then
                Worksheets("Pob_Intermedia").Cells(i, 1).Value = Pos
            Else
                i = i - 1
            End If
        End If
        If i > 2 Then 'cuando selecciona del tercero en adelante
            Pos = Int((Tpob - 1) * Rnd + 1)
            If Pos <> Worksheets("Pob_Intermedia").Cells(i - 1, 1).Value And Pos <>
Worksheets("Pob_Intermedia").Cells(i - 2, 1).Value Then
                Worksheets("Pob_Intermedia").Cells(i, 1).Value = Pos
            Else
                i = i - 1
            End If
        End If
    Next i

End Sub
'.....
'Realiza la recombinacion entre 2 padres y 1 punto
'.....
Public Sub Recombinacion_1pto()

    Dim Tpob, Nact, Nhijos, hijosxpadre, Ngenes, Npadres, cont, cont2, fila, corte, P1, P2, escape As Integer

    Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamano de la poblacion
    Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades
    Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos
    Ngenes = Nact * 2
    hijosxpadre = Worksheets("control").Cells(8, 2).Value 'numero de hijos por padre
    Randomize
    escape = 1
    Worksheets("Pob_Intermedia").Range("B1:IV10000").ClearContents

```

```

While escape < Nhijos + 1 'llenar toda la hoja de calculos de hijos

For cont = 1 To 2 * Nhijos Step 2 'genera hijos
  If (escape > Nhijos) Then
    Exit For
  End If

  P1 = Worksheets("Pob_intermedia").Cells(cont, 1).Value 'elige padre1
  P2 = Worksheets("Pob_intermedia").Cells(cont + 1, 1).Value 'elige padre2

  For cont2 = 1 To hijosxpadre
    If escape > Nhijos Then
      Exit For
    End If

    corte = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un punto de corte que no sera ni el primero ni el ultimo

    If cont2 = 1 Then
      For fila = 3 To (Ngenes + 2)

        If (fila <= (corte + 2)) Then
          Worksheets("Pob_intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
          'Arriba se pasa la informacion del padre 1 al hijo
        Else
          Worksheets("Pob_intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
          'Arriba se pasa la info del padre 2 al hijo
        End If
      Next fila

    ElseIf cont2 = 2 Then

      For fila = 3 To (Ngenes + 2)

        If (fila <= (corte + 2)) Then
          Worksheets("Pob_intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
          'Arriba se pasa la informacion del padre 1 al hijo
        Else
          Worksheets("Pob_intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
          'Arriba se pasa la info del padre 2 al hijo
        End If
      Next fila

    End If

    escape = escape + 1
  Next cont2

Next cont

```

```
Wend
End Sub
```

```
.....
```

```
'Realiza la recombinacion entre 2 padres con 2 puntos
```

```
.....
```

```
Public Sub Recombinacion_2pts_2padres()
```

```
Dim Tpob, Nact, Nhijos, hijosxpadre, Ngenes, escape, cont, cont2, fila, punto1, corte1, punto2, corte2, P1, P2, modo As Integer
```

```
Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamano de la poblacion
```

```
Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades
```

```
Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos
```

```
Ngenes = Nact * 2
```

```
escape = 1
```

```
Randomize
```

```
hijosxpadre = Worksheets("control").Cells(8, 2).Value
```

```
Worksheets("Pob_Intermedia").Range("B1:IV10000").ClearContents
```

```
While escape < Nhijos + 1
```

```
For cont = 1 To 2 * Nhijos Step 2 'genera hijos
```

```
    If escape > Nhijos Then
```

```
        Exit For
```

```
    End If
```

```
    P1 = Worksheets("Pob_intermedia").Cells(cont, 1).Value
```

```
    P2 = Worksheets("Pob_intermedia").Cells(cont + 1, 1).Value
```

```
    For cont2 = 1 To hijosxpadre
```

```
        If escape > Nhijos Then
```

```
            Exit For
```

```
        End If
```

```
        punto1 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un punto de corte que no sera ni el primero ni el ultimo
```

```
        punto2 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un punto de corte que no sera ni el primero ni el ultimo
```

```
        While (punto1 = punto2) 'revisa el caso si los pts de corte son iguales
```

```
            punto1 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un nuevo punto de corte
```

```
        Wend
```

```
        If (punto1 > punto2) Then 'ordena los puntos de corte
```

```
            corte2 = punto1
```

```
            corte1 = punto2
```

```
        Else
```

```
            corte2 = punto2
```

```
            corte1 = punto1
```

```
        End If
```

```
    If (cont2 = 1) Then
```

```

For fila = 3 To (Ngenes + 2)
    If (fila <= (corte1 + 2)) Then
        Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
        'Arriba se pasa la informacion del padre 1 al hijo por el modo 1

        ElseIf (fila <= (corte2 + 2)) Then
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
            'Arriba se pasa la info del padre 2 al hijo por el modo 1
        Else
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
            'Arriba se pasa la informacion del padre 1 al hijo por el modo 1
        End If

    Next fila

    ElseIf cont2 = 2 Then

        For fila = 3 To (Ngenes + 2)

            If (fila <= (corte1 + 2)) Then
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
                'Arriba se pasa la informacion del padre 1 al hijo por el modo 2

            ElseIf (fila <= (corte2 + 2)) Then
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
                'Arriba se pasa la info del padre 2 al hijo por el modo 2
            Else
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
                'Arriba se pasa la informacion del padre 1 al hijo por el modo 2
            End If

        Next fila

    End If
    escape = escape + 1
    Next cont2
Next cont

Wend
End Sub

.....
'Realiza la recombinacion entre 3 padres con 2 puntos
.....
Public Sub Recombinacion_2pts_3padres()

```

Dim Tpob, Nact, cont, hijosxpadre, escape, Nhijos, Ngenes, cont2, fila, punto1, corte1, punto2, corte2, P1, P2, P3, modo As Integer

```
Tpob = Worksheets("Control").Cells(2, 2).Value 'Tamano de la poblacion
Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades
Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos
hijosxpadre = Worksheets("control").Cells(8, 2).Value 'Numero de hijos de cada grupo de padres escojidos
Ngenes = Nact * 2
cont = 0
cont2 = 1
Randomize
escape = 1
Worksheets("Pob_Intermedia").Range("B1:IV10000").ClearContents
```

```
While (escape < Nhijos + 1) 'genera hijos
  For cont = 1 To 3 * Nhijos Step 3
```

```
    If escape > Nhijos Then
      Exit For
    End If
```

```
    P1 = Worksheets("pob_intermedia").Cells(cont, 1).Value 'elige padres
    P2 = Worksheets("pob_intermedia").Cells(cont + 1, 1).Value 'elige padres
    P3 = Worksheets("pob_intermedia").Cells(cont + 2, 1).Value 'elige padres
```

```
    For cont2 = 1 To hijosxpadre 'genera # de hijos por cada grupo de padres especificados
```

```
      If (escape > Nhijos) Then ' revisa que hijosxpadre divisor de # total hijos
        Exit For
      End If
```

```
      punto1 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un punto de corte que no sera ni el primero ni el ultimo
      punto2 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un punto de corte que no sera ni el primero ni el ultimo
```

```
      While (punto1 = punto2) 'revisa el caso si los pts de corte son iguales
        punto1 = Int(((Nact - 1) * Rnd) + 1) * 2 'elige un nuevo punto de corte
      Wend
```

```
      If (punto1 > punto2) Then ' ordena los puntos de corte
        corte2 = punto1
        corte1 = punto2
      Else
        corte2 = punto2
        corte1 = punto1
      End If
```

```
      If (cont2 = 1) Then
        For fila = 3 To (Ngenes + 2)
```

```
          If (fila <= (corte1 + 2)) Then
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
            'Arriba se pasa la informacion del padre 1 al hijo por el modo 1
```

```

        ElseIf (fila <= (corte2 + 2)) Then
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
            'Arriba se pasa la info del padre 2 al hijo por el modo 1
        Else
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P3).Value
            'Arriba se pasa la informacion del padre 3 al hijo por el modo 1
        End If

    Next fila

ElseIf (cont2 = 2) Then

    For fila = 3 To (Ngenes + 2)

        If (fila <= (corte1 + 2)) Then
            Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P3).Value
            'Arriba se pasa la informacion del padre 1 al hijo por el modo 2

            ElseIf (fila <= (corte2 + 2)) Then
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
                'Arriba se pasa la info del padre 3 al hijo por el modo 2
            Else
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
                'Arriba se pasa la informacion del padre 2 al hijo por el modo 2
            End If

        Next fila

    ElseIf (cont2 = 3) Then

        For fila = 3 To (Ngenes + 2)

            If (fila <= (corte1 + 2)) Then
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P2).Value
                'Arriba se pasa la informacion del padre 2 al hijo por el modo 3

            ElseIf (fila <= (corte2 + 2)) Then
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P3).Value
                'Arriba se pasa la info del padre 3 al hijo por el modo 3
            Else
                Worksheets("Pob_Intermedia").Cells(fila, escape + 1).Value =
Worksheets("Población").Cells(fila, P1).Value
                'Arriba se pasa la informacion del padre 1 al hijo por el modo 3
            End If

        Next fila

```

```

End If

    escape = escape + 1
Next cont2
Next cont
Wend

End Sub

'*****
'Realiza mutacion sumando o restando 1 a la posicion inicial
'*****

Public Sub mutacion_incremento_decremento()

Dim Mutacion, Tpob, Nhijos, Nact, cambio, Ngenes, original, nuevo, Generacion, Actividad, Cont_mutacion,
i, columna, fila, Posicion, Rango, Minimo, maximo As Integer

Mutacion = Worksheets("Control").Cells(5, 2).Value 'Procentaje de mutacion
Tpob = Worksheets("control").Cells(2, 2).Value ' Tamano de poblacion
Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos
Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades
Ngenes = Nact * 2 ' Numero de genes

Generacion = Nhijos * Ngenes 'total de casillas en la poblacion de una generacion

Cont_mutacion = Int(Generacion * Mutacion) 'Numero de veces que hay mutacion

Randomize

For i = 1 To Cont_mutacion

    Posicion = Int(Generacion * Rnd) + 1 'elige una posicion para mutar
    columna = Int((Posicion - 1) / Ngenes) + 1 + 1 'encuentra la columna tiene en cuenta columna de padres
    fila = Posicion Mod Ngenes 'encuentra la fila

    If (fila = 0) Then
        fila = fila + Ngenes 'corrige las veces que quede en la ultima fila
    End If

    cambio = Int((Rnd - 0.5))

    If (cambio = 0) Then
        cambio = 1
    End If

    If ((Posicion Mod 2) = 1) Then 'revisa si hay restriccion por duraciones

        Actividad = Int(fila / 2) + 1 ' encuentra cual es la actividad que va a ser mutada
        Minimo = Worksheets("Actividades").Cells((Actividad + 1), 6).Value
        maximo = Worksheets("Actividades").Cells((Actividad + 1), 5).Value
        Rango = maximo - Minimo
        'arriba se encontro el rango en el que se puede hacer mutacion

```



```

original = Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value 'encuentra el valor que
estaba originalmente en la casilla
nuevo = original + cambio 'muta

```

```

If (Rango >= 1) Then 'revisa que no salga del rango

```

```

    If (nuevo >= Minimo And nuevo <= maximo) Then

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = nuevo 'muta

```

```

    ElseIf (nuevo > maximo) Then

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = maximo - 1

```

```

    ElseIf (nuevo < Minimo) Then

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = Minimo + 1

```

```

    End If

```

```

End If

```

```

Else

```

```

    Actividad = Int(fila / 2) + 1

```

```

    original = Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value 'encuentra el valor que
estaba originalmente en la casilla

```

```

    nuevo = original + cambio

```

```

    Rango = Worksheets("Actividades").Cells((Actividad + 1), 7).Value

```

```

    If (nuevo >= 0 And nuevo <= Rango) Then

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = nuevo 'muta

```

```

    ElseIf (nuevo > Rango) Then

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = Rango 'muta

```

```

    Else

```

```

        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = 0 'muta

```

```

    End If

```

```

End If

```

```

Next i

```

```

End Sub

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

'Realiza mutacion cuyo porcentaje varia dependiendo del numero de generacion

```

```

'actual

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

Public Sub Mutacion_variable()

```

```

    Dim Mutacion, minmutacion, generacion_actual, Tpob, Nhijos, Nact, Ngenes, original, nuevo, Generacion,
Actividad, Cont_mutacion, i, columna, fila, Posicion, Rango, Minimo, maximo As Integer

```

```

    Mutacion = Worksheets("Control").Cells(5, 2).Value 'Procentaje de mutacion

```

```

    Tpob = Worksheets("control").Cells(2, 2).Value ' Tamano de poblacion

```

```

Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos
Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades
minmutacion = Worksheets("Control").Cells(10, 2).Value 'mutacion minima
Ngenes = Nact * 2 ' Numero de genes

Generacion = Nhijos * Ngenes 'total de casillas en la poblacion de una generacion
generacion_actual = Worksheets("control").Cells(9, 2).Value 'numero de generacion actual

Cont_mutacion = Int((minmutacion * 100) + (Mutacion * generacion_actual)) 'Numero de veces que hay
mutacion en la generacion actual

Randomize

For i = 1 To Cont_mutacion

    Posicion = Int(Generacion * Rnd) + 1 'elige una posicion para mutar
    columna = Int((Posicion - 1) / Ngenes) + 1 + 1 'encuentra la columna tiene en cuenta columna de padres
    fila = Posicion Mod Ngenes 'encuentra la fila

    If (fila = 0) Then
        fila = fila + Ngenes 'corrige las veces que quede en la ultima fila
    End If

    If ((Posicion Mod 2) = 1) Then 'revisa si hay restriccion por duraciones

        Actividad = Int(fila / 2) + 1 ' encuentra cual es la actividad que va a ser mutada
        Minimo = Worksheets("Actividades").Cells((Actividad + 1), 6).Value
        maximo = Worksheets("Actividades").Cells((Actividad + 1), 5).Value
        Rango = maximo - Minimo
        'arriba se encontro el rango en el que se puede hacer mutacion
        If (Rango >= 1) Then
            original = Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value 'encuentra el valor que
            estaba originalmente en la casilla
            nuevo = Minimo + Int(Rango * Rnd) 'muta
            If (original = nuevo) Then
                nuevo = Minimo + Int(Rango * Rnd) ' revisa que no quede el mismo valor
            End If

            Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = nuevo 'muta

        End If
    Else
        Actividad = Int(fila / 2) + 1
        Rango = Worksheets("Actividades").Cells(Actividad + 1, 7).Value 'define rango para mutar
        Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = Int(Rango * Rnd) 'muta

    End If

Next i

End Sub

```

.....

'Realiza mutacion aleatoria

.....

Public Sub Mutacion_aleatoria()

Dim Mutacion, Tprob, Nhijos, Nact, Ngenes, original, nuevo, Generacion, Actividad, Cont_mutacion, i, columna, fila, Posicion, Rango, Minimo, maximo As Integer

Mutacion = Worksheets("Control").Cells(5, 2).Value 'Procentaje de mutacion

Tprob = Worksheets("control").Cells(2, 2).Value ' Tamano de poblacion

Nhijos = Worksheets("Control").Cells(7, 2).Value 'Numero de hijos

Nact = Worksheets("Control").Cells(1, 2).Value 'Numero de actividades

Ngenes = Nact * 2 ' Numero de genes

Generacion = Nhijos * Ngenes 'total de casillas en la poblacion de una generacion

Cont_mutacion = Int(Generacion * Mutacion) 'Numero de veces que hay mutacion

Randomize

For i = 1 To Cont_mutacion

Posicion = Int(Generacion * Rnd) + 1 'elige una posicion para mutar

columna = Int((Posicion - 1) / Ngenes) + 1 + 1 'encuentra la columna, tiene en cuenta desplazamiento por columna de padres

fila = Posicion Mod Ngenes 'encuentra la fila

If (fila = 0) Then

fila = fila + Ngenes 'corrige las veces que quede en la ultima fila

End If

If ((Posicion Mod 2) = 1) Then 'revisa si hay restriccion por duraciones

Actividad = Int(fila / 2) + 1 ' encuentra cual es la actividad que va a ser mutada

Minimo = Worksheets("Actividades").Cells((Actividad + 1), 6).Value

maximo = Worksheets("Actividades").Cells((Actividad + 1), 5).Value

Rango = maximo - Minimo

'arriba se encontro el rango en el que se puede hacer mutacion

If (Rango >= 1) Then

original = Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value 'encuentra el valor que estaba originalmente en la casilla

nuevo = Minimo + Int(Rango * Rnd) 'muta

If (original = nuevo) Then

nuevo = Minimo + Int(Rango * Rnd) ' revisa que no quede el mismo valor

End If

Worksheets("Pob_intermedia").Cells((fila + 2), columna).Value = nuevo 'muta

End If

Else

Actividad = Int(fila / 2) + 1


```

        If Worksheets("Población").Cells(x, i).Value = Worksheets("Pob_Intermedia").Cells(x, j -
Tprob).Value Then
            aux1 = x - 2
            Else
                Exit For
            End If
        End If
        If i > Tprob + 1 And j > Tprob + 1 Then 'mira si los ind i y j estan en pob intermedia
        If Worksheets("Pob_Intermedia").Cells(x, i - Tprob).Value =
Worksheets("Pob_Intermedia").Cells(x, j - Tprob).Value Then
            aux1 = x - 2
            Else
                Exit For
            End If
        End If
    Next x
    If aux1 = Ngenes Then 'mira si el cromosoma del ind i estaba igual al del ind j
        If j <= Tprob Then 'mira si j esta en poblacion
            Worksheets("Población").Cells(1, j) = -1
        End If
        If j > Tprob + 1 Then 'mira si j esta en pob intermedia
            Worksheets("Pob_Intermedia").Cells(1, j - Tprob) = -1
        End If
    End If
End If
Next j
End If
Next i

'Inicialización vector Elite
For i = 1 To Tprob
    Elite(i) = i
Next i

'Ordenar para escoger los mejores Nind padres de Población
For i = 1 To Nind
    For j = i + 1 To Tprob
        If (Worksheets("Población").Cells(1, Elite(i)) < Worksheets("Población").Cells(1, Elite(j))) Then
            aux = Elite(i): Elite(i) = Elite(j): Elite(j) = aux
        End If
    Next j
Next i

Worksheets("Pob_Mejores").Range("A1:IV10000").ClearContents

'Copiar los mejores padres a la hoja Pob_Mejores
For i = 1 To Nind 'Recorre el vector Elite

    For fila = 1 To (Ngenes + 2) Step 2 'Recorre la hoja Pob_Mejores
        'Copia el individuo en la posicion Elite (i)
        'de la Hoja Poblacion a la Hoja Pob_Intermedia
        Worksheets("Pob_Mejores").Cells(fila, i).Value = Worksheets("Población").Cells(fila, Elite(i)).Value
        Worksheets("Pob_Mejores").Cells(fila + 1, i).Value = Worksheets("Población").Cells(fila + 1,
Elite(i)).Value
    
```

```

    Next fila
Next i

'Inicialización vector Eliteh
For i = 1 To 256
    Eliteh(i) = i
Next i

'Ordenar para escoger los mejores Nhij hijos de Pob_Intermedia
For i = 2 To (Nhij + 1)
    For j = i + 1 To 256
        If (Worksheets("Pob_Intermedia").Cells(1, Eliteh(i)) < Worksheets("Pob_Intermedia").Cells(1,
Eliteh(j))) Then
            aux = Eliteh(i): Eliteh(i) = Eliteh(j): Eliteh(j) = aux
        End If
    Next j
Next i

'Copiar los mejores hijos a la hoja Pob_Mejores
For i = 1 To Nhij 'Recorre el vector Eliteh

    For fila = 1 To (Ngenes + 2) Step 2 'Recorre la hoja Pob_Mejores
        'Copia el individuo en la posición Eliteh(i)
        'de la Hoja Pob_Intermedia a la Hoja Pob_Mejores
        Worksheets("Pob_Mejores").Cells(fila, i + Nind).Value = Worksheets("Pob_Intermedia").Cells(fila,
Eliteh(i + 1)).Value
        Worksheets("Pob_Mejores").Cells(fila + 1, i + Nind).Value =
Worksheets("Pob_Intermedia").Cells(fila + 1, Eliteh(i + 1)).Value
    Next fila
Next i
End Sub

.....
'Ordena los mejores padres e hijos, presentes en Población y Pob_Intermedia y los copia en
'Pob_Mejores
.....
Public Sub Mejores_PadresHijos()

    Dim Tprob, Tvorg, Nact, i, j, x As Integer
    Dim Vorg(), aux, aux1 As Single 'Vector en el que se guardan las posiciones de los individuos

    Nact = Worksheets("Control").Cells(1, 2).Value 'Número Actividades
    Tprob = Worksheets("Control").Cells(2, 2).Value 'Tamaño Poblacion
    Nhijos = Worksheets("Control").Cells(7, 2).Value 'Cantidad de hijos
    Tvorg = Tprob + Nhijos + 1
    Ngenes = Nact * 2

    ReDim Vorg(2, Tvorg)

    'penalizar los hijos identicos entre ellos
    For i = 2 To Nhijos + 1
        For j = i + 1 To Nhijos + 1

```

```

        'If Fix(Worksheets("Pob_Intermedia").Cells(1, i).Value / 10000) =
Fix(Worksheets("Pob_Intermedia").Cells(1, j).Value / 10000) Then
        'aux1 = Worksheets("Pob_Intermedia").Cells(1, i).Value
        'Worksheets("Pob_Intermedia").Cells(1, i).Value = (aux1 / 1000000000)
        'End If
    'Next j
'Next i

'penalizar los hijos identicos a los padres, para evitar convergencia prematura
'For i = 2 To Nhijos + 1
    'For j = 1 To Tprob
        'If Fix(Worksheets("Pob_Intermedia").Cells(1, i).Value / 10000) =
Fix(Worksheets("Población").Cells(1, j).Value / 10000) Then
        'aux1 = Worksheets("Pob_Intermedia").Cells(1, i).Value
        'Worksheets("Pob_Intermedia").Cells(1, i).Value = (aux1 / 1000000000)
        'End If
    'Next j
'Next i

'Penalizar padres identicos entre ellos
'For i = 1 To Tprob
    'For j = i + 1 To Tprob
        'If Fix(Worksheets("Población").Cells(1, i).Value / 10000) = Fix(Worksheets("Población").Cells(1,
j).Value / 10000) Then
        'aux1 = Worksheets("Población").Cells(1, i).Value
        'Worksheets("Población").Cells(1, i).Value = (aux1 / 1000000000)
        'End If
    'Next j
'Next i

'compara cada individuo con todos para ver cual es igual, si hay alguno igual lo penaliza
'For i = 1 To Tprob + Nhijos + 1 'recorre poblacion y poblacion intermedia
    'If i <> Tprob + 1 Then 'mira que no compare la primera columna de pob intermedia
        'For j = i + 1 To Tprob + Nhijos + 1 'recorre poblacion y pob intermedia para comparar el ind i con los
ind j
            'If j <> Tprob + 1 Then 'mira que no compare la primera columna de pob intermedia
                aux1 = 0
                'For x = 3 To Ngenes + 2 'recorre los cromosmas
                    'If i <= Tprob And j <= Tprob Then 'mira si el ind i esta en poblacion y el ind j tambien
                        'If Worksheets("Población").Cells(x, i).Value = Worksheets("Población").Cells(x, j).Value
Then
                            aux1 = x - 2
                        Else
                            Exit For
                        End If
                    End If
                    'If i <= Tprob And j > Tprob + 1 Then 'mira si el ind i esta en poblacion y el ind j en pob
intermdeia
                        'If Worksheets("Población").Cells(x, i).Value = Worksheets("Pob_Intermedia").Cells(x, j -
Tprob).Value Then
                            aux1 = x - 2
                        Else
                            Exit For
                        End If
                    End If
                End For
            End If
        End For
    End If

```

```

        End If
        If i > Tpob + 1 And j > Tpob + 1 Then 'mira si los ind i y j estan en pob intermedia
            If Worksheets("Pob_Intermedia").Cells(x, i - Tpob).Value =
Worksheets("Pob_Intermedia").Cells(x, j - Tpob).Value Then
                aux1 = x - 2
            Else
                Exit For
            End If
        End If
    Next x
    If aux1 = Ngenes Then 'mira si el cromosoma del ind i estaba igual al del ind j
        If j <= Tpob Then 'mira si j esta en poblacion
            Worksheets("Población").Cells(1, j) = -1
        End If
        If j > Tpob + 1 Then 'mira si j esta en pob intermedia
            Worksheets("Pob_Intermedia").Cells(1, j - Tpob) = -1
        End If
    End If
End If
Next j
End If
Next i

```

```

'Inicializa Vorg con la posicion de cada individuo y el valor de aptitud asociado
For i = 1 To Tpob 'llena el vector con los padres en poblacion
    Vorg(1, i) = i
    Vorg(2, i) = Worksheets("Población").Cells(1, i).Value
Next i
For i = 2 To Nhijos + 1 'llena el vector con los hijos en Pob_Intermedia
    Vorg(1, i + Tpob) = i + Tpob
    Vorg(2, i + Tpob) = Worksheets("Pob_Intermedia").Cells(1, i).Value
Next i

```

```

'Ordena los mejores Tpob individuos en Vorg
For i = 1 To Tpob
    If i <> Tpob + 1 Then
        For j = i + 1 To Tvorg
            If j <> Tpob + 1 Then
                If Vorg(2, i) < Vorg(2, j) Then
                    aux = Vorg(2, i): Vorg(2, i) = Vorg(2, j): Vorg(2, j) = aux
                    aux = Vorg(1, i): Vorg(1, i) = Vorg(1, j): Vorg(1, j) = aux
                End If
            End If
        Next j
    End If
Next i

```

```
Worksheets("Pob_Mejores").Range("A1:IV10000").ClearContents
```

```

'Escribe en Pob_Mejores los mejores padres e hijos
For i = 1 To Tpob

```



```

If Vorg(1, i) < Tpob + 1 Then 'Pregunta si esta en la hoja Población
  For j = 1 To Ngenes + 2
    Worksheets("Pob_Mejores").Cells(j, i).Value = Worksheets("Población").Cells(j, Vorg(1, i)).Value
  Next j
End If
If Vorg(1, i) > Tpob + 1 Then 'Pregunta si esta en la hoja Pob_Intermedia
  For j = 1 To Ngenes + 2
    Worksheets("Pob_Mejores").Cells(j, i).Value = Worksheets("Pob_Intermedia").Cells(j, Vorg(1, i) -
Tpob).Value
  Next j
End If

Next i

End Sub

'Calcula el valor de adaptacion de cada individuo en Pob_Inicial
Public Sub adaptacion_pob_inicial()

  'Se arma la matriz listasfs que contiene el inicio y fin de cada actividad

  Dim original(), listasfs(), Nact, Ngenes, Predecesora, Nhijos, sumatoria, escape, cont, fingrande, fintemp,
codigo, todas_predecesoras, predecesoras_actividad As Integer
  Dim Tpob, K, i, j, s, o, l, y, z, generacion_actual, maximo As Integer

  Tpob = Worksheets("Control").Cells(2, 2).Value 'encuentra el tamaño de la población
  Nhijos = Worksheets("Control").Cells(7, 2).Value 'encuentra el número de hijos por generación
  Nact = Worksheets("Control").Cells(1, 2).Value 'encuentra el número de actividades del proyecto
  Ngenes = Nact * 2
  generacion_actual = Worksheets("Control").Cells(9, 2).Value 'revisa cuál es la generación actual

  cont = 0
  Predecesora = 0
  escape = 0

  For i = 1 To Nact

    If (cont > Predecesora) Then
      Predecesora = cont
    End If

    cont = 0

    For j = 1 To 50

      If (Worksheets("Precedencias").Cells(i + 1, j + 1).Value > 0) Then
        cont = cont + 1
      End If
    Next j
  Next i

```

```

' Arriba se encontro cual es el maximo numero de predecesoras que tiene una actividad para armar matriz

ReDim original(Nact, Predecesora + 1) 'se dimensiona matriz donde se encontrara la info original
ReDim listasfs(Tpob, Nact, 3) 'se dimensiona vector de actividades terminadas con su comienzo y fin

For K = 1 To Tpob 'Recorre toda la poblaci3n

  For i = 1 To Nact

    If i = 1 Then 'llena original, esta con if para que solo lo haga en el primer paso del For
      For y = 1 To Nact
        For z = 1 To Predecesora + 1
          original(y, z) = Worksheets("Precedencias").Cells(y + 1, z).Value 'llena la matriz con la
informacion original
        Next z
      Next y
    End If

    If (original(i, 2) = 0) Then ' comienza con las actividades sin predecesoras

      original(i, 1) = 0 'borra actividad lista de la original
      listasfs(K, i, 1) = Worksheets("Precedencias").Cells(i + 1, 1).Value ' pasa el codigo a la tabla listasfs
      listasfs(K, i, 2) = Worksheets("Poblaci3n").Cells((i * 2) + 2, K) + 1 'halla dia inicial
      listasfs(K, i, 3) = Worksheets("Poblaci3n").Cells((i * 2) - 1 + 2, K) + listasfs(K, i, 2) - 1 'halla dia
final

    End If
  Next i

  escape = 0
  While escape = 0 'ahora revisa las actividades con predecesoras

    For s = 1 To Nact 'comienza el ciclo para revisar hasta que se borren todas las act de original()

      fingrande = 0
      predecesoras_actividad = 0
      todas_predecesoras = 0
      For o = 1 To Predecesora

        If (Worksheets("Precedencias").Cells(s + 1, o + 1).Value > 0) Then
          predecesoras_actividad = predecesoras_actividad + 1 'cuenta la cantidad de predecesoras
          'que tiene la actividad
        End If
      Next o

      For j = 1 To predecesoras_actividad
        If (original(s, j) = 0) Then 'revisa que la actividad no este en la matriz listasfs
          Exit For 'sale para buscar la siguiente actividad
        End If

        codigo = original(s, j + 1) 'encuentra el codigo de la predecesora

        For l = 1 To Nact

```

```

If ((codigo - listasfs(K, 1, 1)) = 0) Then 'si la actividad ya esta en listasfs

    fintemp = listasfs(K, 1, 3) ' asigna una fecha de fin temporal
    todas_predecesoras = todas_predecesoras + 1 'suma cada vez que encuentra una predecesora
    'y solo continuara hasta que todas sumen
    If (fingrande < fintemp) Then
        fingrande = fintemp
    End If
End If

Next l

If (fingrande = 0) Then
    Exit For
End If

Next j
If (fingrande <> 0 And todas_predecesoras = predecesoras_actividad) Then 'revisa que todas las
opredecesoras cumplan

    original(s, 1) = 0 'borra actividad lista de la original
    listasfs(K, s, 1) = Worksheets("Precedencias").Cells(s + 1, 1).Value ' pasa el codigo a la tabla
listasfs
    listasfs(K, s, 2) = Worksheets("Población").Cells((s * 2) + 2, K) + 1 + fingrande 'halla dia inicial
    listasfs(K, s, 3) = Worksheets("Población").Cells((s * 2) - 1 + 2, K) + listasfs(K, s, 2) - 1 'halla dia
final
End If

Next s

sumatoria = 0

For i = 1 To Nact
    sumatoria = sumatoria + original(i, 1) 'sumatoria de los codigos
Next i

If (sumatoria = 0) Then
    escape = 1 ' revisa cuando la lista original() esta vacia para salir del ciclo
End If

Wend
Next K

```

.....

```

'Se calcula el costo directo de cada actividad y se guarda en la matriz Costod
    .....
Dim Costod() As Single
Dim V, R, D, Q, U, C, aux, M As Single
Dim TamUnit, Insaux, Actaux As Integer

TamUnit = 0
i = 2
'Recorre la hoja Unitarios para saber cuantas celdas hay ocupadas

```

```

While Worksheets("Unitarios").Cells(i, 1).Value <> 0
    TamUnit = TamUnit + 1
    i = i + 1
Wend

ReDim Costod(Tpob, Nact, 2) 'vector en el que se guardan los costos directos de las actividades

For K = 1 To Tpob 'Recorre toda la población
    i = 0
    For i = 2 To (TamUnit + 1) 'For para recorrer toda la hoja de Unitarios y armar el costo directo de cada
    actividad
        Insaux = Worksheets("Unitarios").Cells(i, 2).Value 'guarda en Insaux el codigo del insumo en la fila i
        Actaux = Worksheets("Unitarios").Cells(i, 1).Value 'guarda en Actaux el codigo de la actividad a la que
        perence el insumo
        j = 2
        While Worksheets("Insumos").Cells(j, 1).Value - Insaux <> 0 'Busca la posicion del Insumo en la hoja
        Insumos y lo apunta con j
            j = j + 1
        Wend

        If Worksheets("Insumos").Cells(j, 5) = "M" Then 'Pregunta si es material
            U = Worksheets("Insumos").Cells(j, 4).Value 'Guarda en U el valor unitario del insumo en la posicion
            j
            C = Worksheets("Unitarios").Cells(i, 3).Value 'Guarda en C la cantidad unitaria de insumo
            Q = Worksheets("Actividades").Cells(Actaux + 1, 4).Value 'Guarda la cantidad a ejecutar de la
            actividad a la que corresponde el insumo
            aux = Costod(K, Actaux, 1): Costod(K, Actaux, 1) = aux + (C * Q * U) 'calcula y guarda
            acumuladamente en Costod el costo directo de la actividad
            End If

            If Worksheets("Insumos").Cells(j, 5) = "MO" Then 'Pregunta si es mano de obra
            V = Worksheets("Insumos").Cells(j, 4).Value 'Guarda en V el valor unitario del insumo en la posicion
            j
            R = Worksheets("Unitarios").Cells(i, 4).Value 'Guarda en R el rendimiento de la mano de obra
            Q = Worksheets("Actividades").Cells(Actaux + 1, 4).Value 'Guarda la cantidad a ejecutar de la
            actividad a la que corresponde el insumo
            D = listasfs(K, Actaux, 3) - listasfs(K, Actaux, 2) + 1 'Calcula la duración de la actividad a la que
            pertenece la mano de obra
            aux = Costod(K, Actaux, 2): Costod(K, Actaux, 2) = aux + (V * R * Q) 'calcula y guarda
            acumuladamente en Costod el costo directo de la actividad
            If CheckBox1.Value = True Then
                M = Int(((R * Q) / (D * 8))) + 1
                If M > 1 Then
                    aux = Costod(K, Actaux, 2): Costod(K, Actaux, 2) = aux + ((M - 1) * 10000) 'Asigna un
                    sobrecosto al tener que tener mas de una cuadrilla
                End If
            End If
        End If
    Next i
Next K

```

.....

.....

'Se calcula el costo total de cada actividad en cada periodo y se guarda en Egresopp

```

.....

Dim Nped, Fam, PIP, PFP, PIA, PFA, Dper, Diaf, Dact, Diasp As Integer
Dim CTA, auepp As Single
Dim Egresopp() As Single

'Se encuentra cual es la ultima fecha de finalización entre todos los individuos de la población
For K = 1 To Tpob
  For i = 1 To Nact 'Busca el día final del proyecto y lo guarda en Fam
    If listasfs(K, i, 3) > Fam Then
      Fam = listasfs(K, i, 3)
    End If
  Next i
Next K

Dper = Worksheets("Control").Cells(11, 2).Value 'Duracion en dias de cada periodo
PFP = (((Fam - 1) \ Dper) + 1) 'Se calcula el maximo numero de periodos en la Población

ReDim Egresopp(Tpob, Nact, PFP)

For K = 1 To Tpob
  For i = 1 To Nact 'Recorre Egresopp verticalmente
    Dact = listasfs(K, i, 3) - listasfs(K, i, 2) + 1
    PIA = (((listasfs(K, i, 2) - 1) \ Dper) + 1) 'calcula cual es el periodo inicial de la actividad
    PFA = (((listasfs(K, i, 3) - 1) \ Dper) + 1) 'calcula cual es el periodo final de la actividad
    CTA = Costod(K, i, 1) + Costod(K, i, 2)
    auepp = 0

    If PIA = PFA Then
      Egresopp(K, i, PIA) = CTA
    Else
      For j = PIA To PFA 'Recorre Egresopp horizontalmente
        Diaf = j * Dper 'calcula el dia final del periodo actual
        If listasfs(K, i, 3) > Diaf Then 'pregunta si todavia no esta en le periodo final
          Diasp = (Diaf - listasfs(K, i, 2) + 1)
          Egresopp(K, i, j) = (CTA / Dact) * Diasp - auepp
          auepp = auepp + Egresopp(K, i, j)
        End If
        If listasfs(K, i, 3) <= Diaf Then 'pregunta si esta en el periodo final
          Egresopp(K, i, j) = (CTA / Dact) * (listasfs(K, i, 3) - ((j - 1) * Dper))
        End If
      Next j
    End If
  Next i
Next K
.....

'Calcula el egreso total en cada periodo y lo guarda en Egresotp
.....

Dim Egresotp(), ETP As Single

ReDim Egresotp(Tpob, PFP)

```

```

For K = 1 To Tprob
  For i = 1 To PFP
    ETP = 0
    For j = 1 To Nact
      ETP = Egresopp(K, j, i) + ETP 'suma el costo de cada actividad j, del individuo k, en el periodo i
    Next j
    If i = 1 Then
      Egresotp(K, i) = ETP + Worksheets("Ingreso").Cells(9, 2).Value + Worksheets("Ingreso").Cells(10,
2).Value + Worksheets("Ingreso").Cells(10, 2).Value
      'guarda en Egresotp el egreso total en el periodo i, del individuo k + el costo de administración
mensual + el costo de los honorarios de diseño + el costo de la licencia
    Else
      Egresotp(K, i) = ETP + Worksheets("Ingreso").Cells(9, 2).Value
      'guarda en Egresotp el egreso total en el periodo i, del individuo k + el costo de administración
mensual
    End If
  Next i
Next K
.....

.....

'-Calcula el ingreso en cada periodo, segun el ritmo de ventas (cuotas iniciales) y segun se hayan
' o no completado las actividades limite para que se de el resto del ingreso.
'-Recalcula el egreso en los periodos en los que se hacen entregas, según el costo del lote por unidad
entregada
'-Asigna penalizaciones a los individuos que no cumplen con los plazos maximos de entrega
.....

  Dim IngVent(), IngEnt(), Ing, Penal() As Single
  Dim x, Pact, CAL, DL, DFA, Pl, PM1, PM2, PM3 As Integer
  Pl = Worksheets("Ingreso").Cells(6, 11).Value
  'Inicializar el vector de penalizaciones en 1
  ReDim Penal(Tprob)
  For i = 1 To Tprob
    Penal(i) = 1
  Next i

  ReDim IngVent(PFP)
  'Guarda los ingresos debidos al ritmo de ventas en IngVent
  x = 9
  While Worksheets("Ingreso").Cells(x, 8).Value <> 0
    Pact = Worksheets("Ingreso").Cells(x, 8).Value
    Ing = Worksheets("Ingreso").Cells(x, 11).Value
    If Pact > 0 Then 'cuando la venta se hace una vez iniciada la construcción (periodo = 1 o mayor)
      For i = Pact To (Pact - 1 + Pl)
        If (Pact - 1 + Pl) > PFP Then 'redimensiona IngVent y Egresotp cuando el dinero ingresa luego del dia
de finalizacion de la ultima actividad de contruccion
          ReDim Preserve IngVent((Pact - 1 + Pl)), Egresotp(Tprob, (Pact - 1 + Pl))
          If (Pact - 1 + Pl) > PM1 Then 'guarda el periodo maximo posible mas grande
            PM1 = (Pact - 1 + Pl)
          End If
        End If
      Next i
      IngVent(i) = IngVent(i) + (Ing / Pl)
    Next i
  End While

```

```

End If
If Pact < 0 And ((Pl + Pact) < 1 Or (Pl + Pact) = 1) Then 'cuando la venta se hace antes de iniciar
construcción y la cuota inicial se paga toda antes de iniciar la construccion
    IngVent(1) = IngVent(1) + Ing
End If
If Pact < 0 And (Pl + Pact) > 1 Then
    If (Pl + Pact) > PFP Then 'redimensiona IngVent y Egresotp cuando el dinero ingresa luego del dia de
finalizacion de la ultima actividad de contruccion
        ReDim Preserve IngVent((Pl + Pact)), Egresotp(Tpob, (Pl + Pact))
        If (Pl + Pact) > PM2 Then
            PM2 = (Pl + Pact) 'guarda el periodo maximo posible mas grande
        End If
    End If
    IngVent(1) = IngVent(1) + ((Ing / Pl) * (Abs(Pact) + 1))
    For i = 2 To (Pl + Pact)
        IngVent(i) = IngVent(i) + (Ing / Pl)
    Next i
End If
x = x + 1
Wend

If PM1 > PM2 Then ' mira cual fue el ultimo periodo de pago del proyecto
    PM3 = PM1
ElseIf PM2 > PM1 Then
    PM3 = PM2
ElseIf PFP > PM3 Then
    PM3 = PFP
End If

ReDim IngEnt(Tpob, PM3)
For K = 1 To Tpob
    'Guarda los ingresos debidos al ritmo de entregas en IngEnt
    x = 9
    While Worksheets("Ingreso").Cells(x, 15).Value <> 0
        CAL = Worksheets("Ingreso").Cells(x, 15).Value 'Codigo de actividad limite
        DL = Worksheets("Ingreso").Cells(x, 16).Value 'Dia limite de entrega
        Ing = Worksheets("Ingreso").Cells(x, 18).Value - Worksheets("Ingreso").Cells(x, 19).Value -
Worksheets("Ingreso").Cells(x, 20).Value - Worksheets("Ingreso").Cells(x, 21).Value
        'El ingreso debido a la entrega de la(s) unidades es igual al valor restante menos el valor del terreno,
menos el costos notarial y menos la comision

        DFA = listasfs(K, CAL, 3) 'Dia de finalizacion de la actividad limite
        PFA = (((DFA - 1) \ Dper) + 1) 'Periodo en el que finaliza la actividad limite

        If DFA > DL Then
            IngEnt(K, PFA) = IngEnt(K, PFA) + Ing
            Penal(K) = Penal(K) / 1000000
        End If
        If DFA <= DL Then
            IngEnt(K, PFA) = IngEnt(K, PFA) + Ing
        End If
        x = x + 1
    Wend
Next K

```

```

.....

.....
'Calculo del valor de adaptación
.....

Dim Adapt(), FCneto(), FCacum(), Mfp(), Mfc, VPN(), TIO As Single

'Flujos de caja siempre positivos
If OptionButton17.Value = True Then
    ReDim FCacum(Tpob, PM3) 'guarda los flujos de caja acumulados
    ReDim Adapt(Tpob) 'guarda los valores de adaptacion
    ReDim Mfp(Tpob) 'guarda el minimo flujo de caja acumulado de cada individuo
    Mfc = FCacum(1, 1 - 1) + IngVent(1) + IngEnt(1, 1) - Egresotp(1, 1) 'inicializa el minimo
    For K = 1 To Tpob
        Mfp(K) = FCacum(K, 1 - 1) + IngVent(1) + IngEnt(K, 1) - Egresotp(K, 1) 'inicializa el minimo
        For j = 1 To PM3
            FCacum(K, j) = FCacum(K, j - 1) + IngVent(j) + IngEnt(K, j) - Egresotp(K, j) 'calcula el flujo de caja
            acumulado para el individuo k en el periodo j
            If FCacum(K, j) < Mfp(K) Then 'compara el menor flujo de caja para el individuo
                Mfp(K) = FCacum(K, j)
            End If
            If FCacum(K, j) < Mfc Then 'compara el menor flujo de caja en la poblacion
                Mfc = FCacum(K, j)
                min = Mfc
            End If
        Next j
    Next K

    For K = 1 To Tpob
        Adapt(K) = (Mfp(K) + Abs(Mfc) + 1) * Penal(K) 'calcula el valor de adaptacion
        Worksheets("Población").Cells(1, K) = Adapt(K) 'escribe el valor de adaptacion
        Worksheets("Población").Cells(2, K) = Penal(K) 'escribe el factor de penalizacion
    Next K
End If

'VPN
If OptionButton18.Value = True Then
    TIO = Worksheets("Ingreso").Cells(12, 2)
    ReDim FCneto(Tpob, PM3) 'guarda los flujos de caja netos
    ReDim Adapt(Tpob) 'guarda los valores de adaptacion
    ReDim Mfp(Tpob) 'guarda el minimo VPN de cada individuo
    ReDim VPN(Tpob) 'guarda el vpn de cada individuo

    'calcula el VPN del primer individuo para hacerlo igual a Mfc, y poder comparar Mfc con el resto de VPNs
    For i = 1 To PM3
        FCneto(1, i) = IngVent(i) + IngEnt(1, i) - Egresotp(1, i)
        VPN(1) = VPN(1) + (FCneto(1, i) / ((1 + TIO) ^ i))
    Next i
    Mfc = VPN(1)

    'calcula el VPN para el resto de individuos y busca el menor VPN de todos para almacenarlo en Mfc
    If Tpob > 1 Then
        For K = 2 To Tpob

```



```

For i = 1 To PM3
    FCneto(K, i) = IngVent(i) + IngEnt(K, i) + -Egresotp(K, i)
    VPN(K) = VPN(K) + (FCneto(K, i) / ((1 + TIO) ^ i))
Next i
If VPN(K) < Mfc Then
    Mfc = VPN(K)
    min = Mfc
End If
Next K
End If

For K = 1 To Tprob
    Adapt(K) = (VPN(K) + Abs(Mfc) + 1) * Penal(K)
    Worksheets("Población").Cells(1, K) = Adapt(K) 'escribe el valor de adaptacion
    Worksheets("Población").Cells(2, K) = Penal(K) 'escribe el factor de penalizacion
Next K
End If
End Sub

'Calcula el valor de adaptacion de cada individuo en Pob_Intermedia
Public Sub adaptacion_pob_intermedia()

    'Se arma la matriz listasfs que contiene el inicio y fin de cada actividad

    Dim original(), listasfs(), Nact, Ngenes, Predecesora, Nhijos, sumatoria, escape, cont, fnggrande, fintemp,
    codigo, todas_predecesoras, predecesoras_actividad As Integer
    Dim Tprob, K, i, j, s, o, l, y, z, generacion_actual, maximo As Integer

    Tprob = Worksheets("Control").Cells(2, 2).Value 'encuentra el tamaño de la población
    Nhijos = Worksheets("Control").Cells(7, 2).Value 'encuentra el número de hijos por generación
    Nact = Worksheets("Control").Cells(1, 2).Value 'encuentra el número de actividades del proyecto
    Ngenes = Nact * 2
    generacion_actual = Worksheets("Control").Cells(9, 2).Value 'revisa cuál es la generación actual

    cont = 0
    Predecesora = 0
    escape = 0

    For i = 1 To Nact

        If (cont > Predecesora) Then
            Predecesora = cont
        End If

        cont = 0

        For j = 1 To 50

            If (Worksheets("Precedencias").Cells(i + 1, j + 1).Value > 0) Then
                cont = cont + 1
            End If
        Next j
    Next i
End Sub

```

```

        End If
    Next j
Next i

' Arriba se encontro cual es el maximo numero de predecesoras que tiene una actividad para armar matriz

ReDim original(Nact, Predecesora + 1) 'se dimensiona matriz donde se encontrara la info original
ReDim listasfs(Nhijos, Nact, 3) 'se dimensiona vector de actividades terminadas con su comienzo y fin

For K = 1 To Nhijos 'Recorre toda la población

    For y = 1 To Nact
        For z = 1 To Predecesora + 1
            original(y, z) = Worksheets("Precedencias").Cells(y + 1, z).Value 'llena la matriz con la
informacion original
        Next z
    Next y

    For i = 1 To Nact

        If (original(i, 2) = 0) Then ' comienza con las actividades sin predecesoras

            original(i, 1) = 0 'borra actividad lista de la original
            listasfs(K, i, 1) = Worksheets("Precedencias").Cells(i + 1, 1).Value ' pasa el codigo a la tabla listasfs
            listasfs(K, i, 2) = Worksheets("Pob_Intermedia").Cells((i * 2) + 2, K + 1) + 1 'halla dia inicial
            listasfs(K, i, 3) = Worksheets("Pob_Intermedia").Cells((i * 2) - 1 + 2, K + 1) + listasfs(K, i, 2) - 1
'halla dia final

        End If
    Next i

    escape = 0
    While escape = 0 'ahora revisa las actividades con predecesoras

        For s = 1 To Nact 'comienza el ciclo para revisar hasta que se borren todas las act de original()

            fingrande = 0
            predecesoras_actividad = 0
            todas_predecesoras = 0
            For o = 1 To Predecesora

                If (Worksheets("Precedencias").Cells(s + 1, o + 1).Value > 0) Then
                    predecesoras_actividad = predecesoras_actividad + 1 'cuenta la cantidad de predecesoras
                    'que tiene la actividad

                End If
            Next o

            For j = 1 To predecesoras_actividad
                If (original(s, j) = 0) Then 'revisa que la actividad no este en la matriz listasfs
                    Exit For 'sale para buscar la siguiente actividad
                End If

                codigo = original(s, j + 1) 'encuentra el codigo de la predecesora
            
```

```

For l = 1 To Nact
    If ((codigo - listasfs(K, l, 1)) = 0) Then 'si la actividad ya esta en listasfs
        fintemp = listasfs(K, l, 3) ' asigna una fecha de fin temporal
        todas_predecesoras = todas_predecesoras + 1 'suma cada vez que encuentra una predecesora
            'y solo continuara hasta que todas sumen
        If (fingrande < fintemp) Then
            fingrande = fintemp
        End If
    End If

    Next l

    If (fingrande = 0) Then
        Exit For
    End If

    Next j
    If (fingrande <> 0 And todas_predecesoras = predecesoras_actividad) Then 'revisa que todas las
opredecesoras cumplan

        original(s, 1) = 0 'borra actividad lista de la original
        listasfs(K, s, 1) = Worksheets("Precedencias").Cells(s + 1, 1).Value ' pasa el codigo a la tabla
listasfs
        listasfs(K, s, 2) = Worksheets("Pob_Intermedia").Cells((s * 2) + 2, K + 1) + 1 + fingrande 'halla dia
inicial
        listasfs(K, s, 3) = Worksheets("Pob_Intermedia").Cells((s * 2) - 1 + 2, K + 1) + listasfs(K, s, 2) - 1
'halla dia final
    End If

    Next s

    sumatoria = 0

    For i = 1 To Nact
        sumatoria = sumatoria + original(i, 1) 'sumatoria de los codigos
    Next i

    If (sumatoria = 0) Then
        escape = 1 ' revisa cuando la lista original() esta vacia para salir del ciclo
    End If
Wend
Next K
.....

.....
'Se calcula el costo directo de cada actividad y se guarda en la matriz Costod
.....

Dim Costod() As Single
Dim V, R, D, Q, U, C, aux, M As Single
Dim TamUnit, Insaux, Actaux As Integer

TamUnit = 0

```

```

i = 2
'Recorre la hoja Unitarios para saber cuantas celdas hay ocupadas
While Worksheets("Unitarios").Cells(i, 1).Value <> 0
    TamUnit = TamUnit + 1
    i = i + 1
Wend

ReDim Costod(Nhijos, Nact, 2) 'vector en el que se guardan los costos directos de las actividades

For K = 1 To Nhijos 'Recorre toda la población
    i = 0
    For i = 2 To (TamUnit + 1) 'For para recorrer toda la hoja de Unitarios y armar el costo directo de cada
    actividad
        Insaux = Worksheets("Unitarios").Cells(i, 2).Value 'guarda en Insaux el codigo del insumo en la fila i
        Actaux = Worksheets("Unitarios").Cells(i, 1).Value 'guarda en Actaux el codigo de la actividad a la que
        pertenece el insumo
        j = 2
        While Worksheets("Insumos").Cells(j, 1).Value - Insaux <> 0 'Busca la posicion del Insumo en la hoja
        Insumos y lo apunta con j
            j = j + 1
        Wend

        If Worksheets("Insumos").Cells(j, 5) = "M" Then 'Pregunta si es material
            U = Worksheets("Insumos").Cells(j, 4).Value 'Guarda en U el valor unitario del insumo en la posicion
            j
            C = Worksheets("Unitarios").Cells(i, 3).Value 'Guarda en C la cantidad unitaria de insumo
            Q = Worksheets("Actividades").Cells(Actaux + 1, 4).Value 'Guarda la cantidad a ejecutar de la
            actividad a la que corresponde el insumo
            aux = Costod(K, Actaux, 1): Costod(K, Actaux, 1) = aux + (C * Q * U) 'calcula y guarda
            acumuladamente en Costod el costo directo de la actividad
            End If

            If Worksheets("Insumos").Cells(j, 5) = "MO" Then 'Pregunta si es mano de obra
            j
            R = Worksheets("Unitarios").Cells(i, 4).Value 'Guarda en R el rendimiento de la mano de obra
            Q = Worksheets("Actividades").Cells(Actaux + 1, 4).Value 'Guarda la cantidad a ejecutar de la
            actividad a la que corresponde el insumo
            D = listasfs(K, Actaux, 3) - listasfs(K, Actaux, 2) + 1 'Calcula la duración de la actividad a la que
            pertenece la mano de obra
            aux = Costod(K, Actaux, 2): Costod(K, Actaux, 2) = aux + (V * R * Q) 'calcula y guarda
            acumuladamente en Costod el costo directo de la actividad
            If CheckBox1.Value = True Then
                M = Int(((R * Q) / (D * 8))) + 1
                If M > 1 Then
                    aux = Costod(K, Actaux, 2): Costod(K, Actaux, 2) = aux + ((M - 1) * 10000) 'Asigna un
                    sobrecosto al tener mas de una cuadrilla
                End If
            End If
        End If
    Next i
Next K

```

.....

```

.....
'Se calcula el costo total de cada actividad en cada periodo y se guarda en Egresopp
.....

Dim Nped, Fam, PIP, PFP, PIA, PFA, Dper, Diaf, Dact, Diasp As Integer
Dim CTA, auepp As Single
Dim Egresopp() As Single

'Se encuentra cual es la ultima fecha de finalizaci3n entre todos los individuos de la poblaci3n
For K = 1 To Nhijos
  For i = 1 To Nact 'Busca el dia final del proyecto y lo guarda en Fam
    If listasfs(K, i, 3) > Fam Then
      Fam = listasfs(K, i, 3)
    End If
  Next i
Next K

Dper = Worksheets("Control").Cells(11, 2).Value 'Duracion en dias de cada periodo
PFP = (((Fam - 1) \ Dper) + 1) 'Se calcula el maximo numero de periodos en la Poblaci3n

ReDim Egresopp(Nhijos, Nact, PFP)

For K = 1 To Nhijos
  For i = 1 To Nact 'Recorre Egresopp verticalmente
    Dact = listasfs(K, i, 3) - listasfs(K, i, 2) + 1
    PIA = (((listasfs(K, i, 2) - 1) \ Dper) + 1) 'calcula cual es el periodo inicial de la actividad
    PFA = (((listasfs(K, i, 3) - 1) \ Dper) + 1) 'calcula cual es el periodo final de la actividad
    CTA = Costod(K, i, 1) + Costod(K, i, 2)
    auepp = 0

    If PIA = PFA Then
      Egresopp(K, i, PIA) = CTA
    Else
      For j = PIA To PFA 'Recorre Egresopp horizontalmente
        Diaf = j * Dper 'calcula el dia final del periodo actual
        If listasfs(K, i, 3) > Diaf Then 'pregunta si todavia no esta en le periodo final
          Diasp = (Diaf - listasfs(K, i, 2) + 1)
          Egresopp(K, i, j) = (CTA / Dact) * Diasp - auepp
          auepp = auepp + Egresopp(K, i, j)
        End If
        If listasfs(K, i, 3) <= Diaf Then 'pregunta si esta en el periodo final
          Egresopp(K, i, j) = (CTA / Dact) * (listasfs(K, i, 3) - ((j - 1) * Dper))
        End If
      Next j
    End If
  Next i
Next K
.....

.....
'Calcula el egreso total en cada periodo y lo guarda en Egresotp
.....

Dim Egresotp(), ETP As Single

```

```

ReDim Egresotp(Nhijos, PFP)

For K = 1 To Nhijos
  For i = 1 To PFP
    ETP = 0
    For j = 1 To Nact
      ETP = Egresopp(K, j, i) + ETP 'suma el costo de cada actividad j, del individuo k, en el periodo i
    Next j
    If i = 1 Then
      Egresotp(K, i) = ETP + Worksheets("Ingreso").Cells(9, 2).Value + Worksheets("Ingreso").Cells(10,
2).Value + Worksheets("Ingreso").Cells(10, 2).Value
      'guarda en Egresotp el egreso total en el periodo i, del individuo k + el costo de administración
mensual + el costo de los honorarios de diseño + el costo de la licencia
    Else
      Egresotp(K, i) = ETP + Worksheets("Ingreso").Cells(9, 2).Value
      'guarda en Egresotp el egreso total en el periodo i, del individuo k + el costo de administración
mensual
    End If
  Next i
Next K
.....

.....

'-Calcula el ingreso en cada periodo, segun el ritmo de ventas (cuotas iniciales) y segun se hayan
' o no completado las actividades limite para que se de el resto del ingreso.
'-Recalcula el egreso en los periodos en los que se hacen entregas, según el costo del lote por unidad
entregada
'-Asigna penalizaciones a los individuos que no cumplen con los plazos maximos de entrega
.....

Dim IngVent(), IngEnt(), Ing, Penal() As Single
Dim x, Pact, CAL, DL, DFA, Pl, PM1, PM2, PM3 As Integer
Pl = Worksheets("Ingreso").Cells(6, 11).Value

PM1 = 0: PM2 = 0: PM3 = 0

ReDim IngVent(PFP)
'Guarda los ingresos debidos al ritmo de ventas en IngVent
x = 9
While Worksheets("Ingreso").Cells(x, 8).Value <> 0
  Pact = Worksheets("Ingreso").Cells(x, 8).Value
  Ing = Worksheets("Ingreso").Cells(x, 11).Value
  If Pact > 0 Then 'cuando la venta se hace una vez iniciada la construcción (periodo = 1 o mayor)
    For i = Pact To (Pact - 1 + Pl)
      If (Pact - 1 + Pl) > PFP Then 'redimensiona IngVent y Egresotp cuando el dinero ingresa luego del dia
de finalizacion de la ultima actividad de contruccion
        ReDim Preserve IngVent((Pact - 1 + Pl)), Egresotp(Nhijos, (Pact - 1 + Pl))
        If (Pact - 1 + Pl) > PM1 Then 'guarda el periodo maximo posible mas grande
          PM1 = (Pact - 1 + Pl)
        End If
      End If
    Next i
  End If
  IngVent(i) = IngVent(i) + (Ing / Pl)
Next i
End If

```

```

If Pact < 0 And ((Pl + Pact) < 1 Or (Pl + Pact) = 1) Then 'cuando la venta se hace antes de iniciar
construcción y la cuota inicial se paga toda antes de iniciar la construccion
    IngVent(1) = IngVent(1) + Ing
End If
If Pact < 0 And (Pl + Pact) > 1 Then
    IngVent(1) = IngVent(1) + ((Ing / Pl) * (Abs(Pact) + 1))
    For i = 2 To (Pl + Pact)
        If (Pl + Pact) > PFP Then 'redimensiona IngVent y Egresotp cuando el dinero ingresa luego del dia de
finalizacion de la ultima actividad de construccion
            ReDim Preserve IngVent((Pl + Pact)), Egresotp(Nhijos, (Pl + Pact))
            If (Pl + Pact) > PM2 Then
                PM2 = (Pl + Pact) 'guarda el periodo maximo posible mas grande
            End If
        End If
        IngVent(i) = IngVent(i) + (Ing / Pl)
    Next i
End If
x = x + 1
Wend

If PM1 > PM2 Then ' mira cual fue el ultimo periodo de pago del proyecto
    PM3 = PM1
ElseIf PM2 > PM1 Then
    PM3 = PM2
ElseIf PFP > PM3 Then
    PM3 = PFP
End If

'Inicializar el vector de penalizaciones en 1
ReDim Penal(Nhijos)
For i = 1 To Nhijos
    Penal(i) = 1
Next i

ReDim IngEnt(Nhijos, PM3)
For K = 1 To Nhijos
    'Guarda los ingresos debidos al ritmo de entregas en IngEnt
    x = 9
    While Worksheets("Ingreso").Cells(x, 15).Value <> 0
        CAL = Worksheets("Ingreso").Cells(x, 15).Value 'Codigo de actividad limite
        DL = Worksheets("Ingreso").Cells(x, 16).Value 'Dia limite de entrega
        Ing = Worksheets("Ingreso").Cells(x, 18).Value - Worksheets("Ingreso").Cells(x, 19).Value -
Worksheets("Ingreso").Cells(x, 20).Value - Worksheets("Ingreso").Cells(x, 21).Value
        'El ingreso debido a la entrega de la(s) unidades es igual al valor restante menos el valor del terreno,
menos el costos notarial y menos la comision

        DFA = listasfs(K, CAL, 3) 'Dia de finalizacion de la actividad limite
        PFA = (((DFA - 1) \ Dper) + 1) 'Periodo en el que finaliza la actividad limite

        If DFA > DL Then
            IngEnt(K, PFA) = IngEnt(K, PFA) + Ing
            Penal(K) = Penal(K) / 1000000
        End If
        If DFA <= DL Then

```

```

        IngEnt(K, PFA) = IngEnt(K, PFA) + Ing
    End If
    x = x + 1
Wend
Next K
.....

.....

'Calculo del valor de adaptación
.....

Dim Adapt(), FCneto(), FCacum(), Mfp(), Mfc, VPN(), TIO As Single

'Flujos de caja siempre positivos
If OptionButton17.Value = True Then
    ReDim FCacum(Nhijos, PM3) 'guarda los flujos de caja acumulados
    ReDim Adapt(Nhijos) 'guarda los valores de adaptacion
    ReDim Mfp(Nhijos) 'guarda el minimo flujo de caja acumulado de cada individuo
    'Mfc = FCacum(1, 1 - 1) + IngVent(1) + IngEnt(1, 1) - Egresotp(1, 1) 'inicializa el minimo
    Mfc = min
    For K = 1 To Nhijos
        Mfp(K) = FCacum(K, 1 - 1) + IngVent(1) + IngEnt(K, 1) - Egresotp(K, 1) 'inicializa el minimo
        For j = 1 To PM3
            FCacum(K, j) = FCacum(K, j - 1) + IngVent(j) + IngEnt(K, j) - Egresotp(K, j) 'calcula el flujo de caja
            acumulado para el individuo k en el periodo j
            If FCacum(K, j) < Mfp(K) Then 'compara el menor flujo de caja para el individuo
                Mfp(K) = FCacum(K, j)
            End If
            'If FCacum(K, j) < Mfc Then 'compara el menor flujo de caja en la poblacion
            'Mfc = FCacum(K, j)
            'End If 'para que el minimo siempre sea el minimo de los padres
        Next j
    Next K

    For K = 1 To Nhijos
        Adapt(K) = (Mfp(K) + Abs(Mfc) + 1) * Penal(K) 'calcula el valor de adaptacion
        If Adapt(K) < 0 Then
            Adapt(K) = 0 'por si se genera un hijo mas malo que todos los padres, su aptitud sea cero
        End If
        Worksheets("Pob_Intermedia").Cells(1, K + 1) = Adapt(K) 'escribe el valor de adaptacion
        Worksheets("Pob_Intermedia").Cells(2, K + 1) = Penal(K) 'escribe el factor de penalizacion
    Next K
End If

'VPN
If OptionButton18.Value = True Then
    TIO = Worksheets("Ingreso").Cells(12, 2)
    ReDim FCneto(Nhijos, PM3) 'guarda los flujos de caja netos
    ReDim Adapt(Nhijos) 'guarda los valores de adaptacion
    ReDim Mfp(Nhijos) 'guarda el minimo VPN de cada individuo
    ReDim VPN(Nhijos) 'guarda el vpn de cada individuo

    'calcula el VPN del primer individuo para hacerlo igual a Mfc, y poder comparar Mfc con el resto de VPNs
    'For i = 1 To PM3

```



```

    'FCneto(1, i) = IngVent(1) + IngEnt(1, i) + -Egresotp(1, i)
    'VPN(1) = VPN(1) + (FCneto(1, i) / ((1 + TIO) ^ i))
'Next i
'Mfc = VPN(1)
'VPN(1) = 0
Mfc = min

'calcula el VPN para el resto de individuos y busca el menor VPN de todos para almacenarlo en Mfc
If Nhijos > 1 Then
    For K = 1 To Nhijos
        For i = 1 To PM3
            FCneto(K, i) = IngVent(i) + IngEnt(K, i) + -Egresotp(K, i)
            VPN(K) = VPN(K) + (FCneto(K, i) / ((1 + TIO) ^ i))
        Next i
        'If VPN(K) < Mfc Then
        'Mfc = VPN(K)
        'End If          'para que el minimo siempre sea el mas malo de los padres, no el mas malo de los
hijos
    Next K
End If

For K = 1 To Nhijos
    Adapt(K) = (VPN(K) + Abs(Mfc) + 1) * Penal(K)
    If Adapt(K) < 0 Then
        Adapt(K) = 1 'por si se genera un hijo mas malo que todos los padres, su valor no quede negativo
    End If
    Worksheets("Pob_Intermedia").Cells(1, K + 1) = Adapt(K) 'escribe el valor de adaptacion
    Worksheets("Pob_Intermedia").Cells(2, K + 1) = Penal(K) 'escribe el factor de penalizacion
Next K
End If
End Sub

```