Universidad de las Ciencias Informáticas
Laboratorio de Gestión de Proyectos, Facultad 5

Algoritmo genético para optimizar la planificación del cronograma de un proyecto de software

Trabajo final presentado en opción al título de Máster en Gestión de Proyectos Informáticos

Autor: Ing. Susej Beovides Luis

Tutor: Dr.C Mario González Arencibia

La Habana, 2015

"Tras escalar una gran colina, uno solo encuentra que hay muchas más colinas que escalar... siempre parece imposible hasta que se hace."

Nelson Mandela.

Dedicatoria

"A mi familia y amigos cercanos que siempre confiaron en que podía alcanzar este triunfo. A ellos va dedicado este logro en mi vida..."

Agradecimientos

Le agradezco a todos los que pusieron un granito de arena en este trabajo...
A mi familia que siempre ha sabido apoyar mis metas, aún en la distancia...
A mis buenos amigos por soportarme y darme ánimo todas las veces que lo necesité, que no fueron pocas!!!...
Y en especial al profe Pedro por sus regaños oportunos, sus revisiones y sabios consejos...
A todos de corazón muchas gracias!!!

Declaración de autoría

Declaro por este medio que yo Susej Beovides Luis con carné de identidad 85051912013 soy el autor principal del trabajo final de maestría ALGORITMO GENÉTICO PARA OPTIMIZAR LA PLANIFICACIÓN DEL CRONOGRAMA DE UN PROYECTO DE SOFTWARE, desarrolladoo como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste firmo la presente declaración jurada de autoría en La Habana a los ____ días del mes de _____ del año _____.

Ing. Susej Beovides Luis
Autor

Resumen

La adecuada gestión de los proyectos, se ha convertido en un elemento de vital importancia para que estos se conviertan en proyectos exitosos, en tiempo y con calidad. Por esta razón al problema de la planificación de proyectos de software se le presta gran atención en la actualidad y numerosas investigaciones han sido encaminadas a la representación y solución computacional de este problema.

En la Universidad de las Ciencias Informáticas existen varios centros dedicados al desarrollo de software, los cuales gestionan sus productos a través de proyectos. En la planificación de los mismos se utiliza GESPRO, una plataforma orientada a la coordinación, seguimiento de las tareas y a la comunicación entre los participantes del proyecto. Pero la planificación se realiza de forma manual y se evidencian deficiencias en la misma que provocan que la planificación del cronograma de ejecución del proyecto no sea completamente acertada.

Este trabajo tiene la finalidad de automatizar la planificación del cronograma de un proyecto de software a través de un algoritmo genético que minimice el tiempo de desarrollo y optimice la asignación de recursos en el mismo. Se presentan además los resultados del desempeño del algoritmo propuesto mediante instancias de la biblioteca PSPLib de proyectos con 30, 60 y 120 tareas respectivamente. Como conclusión se plantea que el Algoritmo Genético diseñado obtiene soluciones cuasi óptimas para el Problema de Planificación de Proyectos con Recursos Limitados. Además, se logra un adecuado balance de carga entre los recursos de un mismo tipo.

Palabras clave: Algoritmo Genético, Planificación de Proyectos de Software, Tiempo de desarrollo.

Índice general

Índice de figuras

Índice de tablas

Lista de algoritmos

Lista de códigos fuentes

Introducción

En la actualidad la industria del software se encuentra dentro del número creciente de empresas y organizaciones que están adaptando e incorporando la Gestión de Proyectos a sus actividades cotidianas, debido a las ventajas que aporta en ahorros de tiempo, materiales y recursos humanos, aumentando la calidad de los productos, procesos, etc. Esto se logra mediante la aplicación e integración de los procesos de inicio, planificación, ejecución, seguimiento y control, y cierre (PMI, 2013).

El entorno actual de desarrollo de proyectos de software se caracteriza por cambios tecnológicos acelerados, presiones económicas, trabajo en equipos multidisciplinarios, recursos y tiempo limitado. Es por ello que todos los proyectos requieren coordinación, planificación y control de los recursos que intervienen en el mismo, para lograr una mayor calidad del producto y minimizar los costos, el tiempo y los recursos utilizados (INFANTE, 2008).

La gestión de un proyecto de software comienza con un conjunto de actividades que globalmente se denomina planificación del proyecto. Uno de los aspectos más importantes a la hora de planificar un proyecto, consiste en la elaboración de una programación que permita establecer un guión temporal de las acciones y tareas a desarrollar durante su ejecución y que tenga en cuenta todos aquellos factores decisivos a la hora de realizar los trabajos, desde el orden necesario de los mismos hasta la disponibilidad de los recursos. Es muy difícil que un equipo de gestores, ingenieros, técnicos y científicos logre hacer una programación perfecta de un proyecto desde su primera versión, pues la mayoría de las veces deben tener en cuenta un elevado número de alternativas que no pueden evaluar. Por tanto en la medida que el proyecto avanza, es normal que algunas actividades o tareas requieran más o menos tiempo, en dependencia por ejemplo, que la disponibilidad de recursos no sea la prevista, lo cual haga cambiar la planificación de alguna tarea.

En la medida que aumenta la complejidad de un proyecto de software, se hace más imperiosa la necesidad de planificar. La planificación garantiza, en gran medida, la eficiencia en el desempeño del equipo de proyectos, pues establece métodos de utilización racional de los recursos y del tiempo; proporciona los elementos necesarios para llevar a cabo el control y establecer un sistema adecuado para la toma de decisiones que hagan frente a las contingencias que pudieran presentarse.

La combinación de todos estos aspectos y las limitaciones de los métodos tradicionales, hacen del proceso de planificación una tarea difícil (KUMARI y SRINIVAS, 2013), captando el interés de muchos investigadores de diferentes comunidades de investigación, por ejemplo, la Investigación de Operaciones (OR) e Inteligencia Artificial (AI), además a que se tome interés por la representación de las características, en términos de habilidades y conocimientos de los recursos humanos, en los modelos de optimización propuestos

(T. K. ABDEL-HAMID y MADNICK, 1989), (T. ABDEL-HAMID, 1989). (J. FRANCISCO CHICANO, 2007), (CHANG; JIANG; DI; ZHU y GE, 2008), (DONGWON KANG y BAE, 2011).

Las características de la Universidad de las Ciencias Informáticas (UCI) como centro formador y a la vez productor la han distinguido desde sus inicios. Pero esto ha llevado a que se complejicen muchos procesos dentro de la universidad. Teniendo en cuenta el tamaño de la organización y el volumen de datos que se maneja, fue necesario el uso de herramientas informáticas que ayuden a la implantación de un modelo de desarrollo tecnológico y apoyen la toma de decisiones a diferentes niveles (persona, proyecto, centro y alta gerencia) (PÉREZ, 2010). La UCI ha asumido la Suite de Gestión de Proyectos (GESPRO) como sistema base para la gestión de proyectos en su red de centros de desarrollo de soluciones y servicios informáticos. Gestor y planificador con interfaz web, orientado a la coordinación, seguimiento de las tareas y a la comunicación entre los participantes del proyecto. En dicho sistema se realiza la planificación del cronograma de un proyecto pero el mismo presenta limitaciones como son:

- Al planificar una tarea se debe tener conocimiento de las demás tareas que pueden depender de su realización, además de las que deben haber finalizado para que esta pueda iniciarse.
- Los tiempos de inicio y los tiempos de fin se asignan manualmente y se puede violar involuntariamente el principio anterior.
- La asociación del tipo de tarea con el rol del recurso que debe realizarla puede ser violada.
- La asignación de recursos a las tareas puede violar el balance de carga de los mismos al no tener en cuenta la previa asignación de otras tareas.
- Pueden utilizarse para la planificación más cantidad de recursos que los necesarios y viceversa.
- El proceso de planificación es altamente dependiente del conocimiento del planificador del proyecto.

Esta situación provoca deficiencias a la hora de definir una estrategia óptima para el proceso de planificación. A su vez dificulta la toma de decisiones respecto a la ejecución del proyecto debido a su rentabilidad y aplicabilidad.

En general el proceso de planificación actual se torna engorroso por su complejidad de aplicación y se evidencia falta de aprovechamiento de las experiencias en los procesos de planificación anteriores, y ante la pérdida de planificadores por envejecimiento o migración de puestos de trabajo que hace que se pierda el conocimiento adquirido, se evidencia la necesidad de algoritmos de planificación que sean capaces de ofrecer resultados entendibles e interpretables.

Al ser GESPRO una plataforma muy usada por todos los proyectos y extensible, se impone la búsqueda de una solución para la problemática descrita anteriormente, que conlleva al planteamiento del siguiente ***problema científico*** ¿Cómo obtener el cronograma de un proyecto de software de modo tal que, respetando las dependencias entre las tareas, se asignen los recursos mínimos y la duración total estimada del proyecto en el menor tiempo posible?

El problema anterior se enmarca dentro del ***objeto de estudio*** los métodos de optimización aplicados al Problema de Planificación de Proyectos con Recursos Limitados y en la solución del mismo se traza como ***objetivo general*** Desarrollar un algoritmo genético para optimizar la planificación del cronograma de un proyecto de software, garantizando una eficiente asignación de recursos y la duración mínima de la planificación estimada del proyecto.

Para darle cumplimiento al objetivo general, se definen los siguientes ***objetivos específicos***:

- Elaborar el marco teórico de la investigación, identificando el problema de planificación de proyecto que pudiera ser empleado para modelar el proceso de gestión de tiempo de un proyecto de software y representar las decisiones tácticas y operativas en un modelo formal como un problema de optimización.
- Diseñar e implementar un algoritmo genético que construya el cronograma en función de las decisiones representadas.
- Utilizar instancias de prueba, para realizar un estudio experimental de la solución que se propone.

Mientras el **campo de acción** se centra en los métodos de optimización basados en Algoritmos Genéticos (por sus siglas en inglés, Genetic Algorithms) (GA) aplicados al Problema de Planificación de Proyectos con Recursos Limitados.

Tipo de investigación

Descriptiva: Se realiza una descripción detallada de los elementos característicos de los RCPSP, así como las tendencias actuales de solución a estos problemas. Proponiéndose un algoritmo genético para optimizar la planificación del cronograma de un proyecto de software.

Hipótesis

El desarrollo de un algoritmo genético que optimice el cronograma de un proyecto de software con recursos limitados respetando las dependencias entre las tareas y asignando los recursos de forma eficiente para darles cumplimiento a las mismas, permitirá aumentar la eficiencia en el proceso de planificación estimada del cronograma de un proyecto de software.

Operacionalización de las variables

Esta tesis trabaja fundamentalmente con dos variables.

Tabla 1. **Operacionalización de las variables**

Variable independiente: algoritmo genético para la planificación del cronograma de un proyecto de software.

Dimensión	**Indicadores**	**Unidad de Medida**
Planificación del cronograma a través del algoritmo genético	Mejorar	Es correcto
		No es correcto
	Comparación con otros algoritmos	unidades de tiempo.
Costos de desarrollo	Esfuerzo Dedicado	Horas-Hombre
Aplicabilidad	Aplicación de la Propuesta en un caso de estudio	Si
		No

Variable dependiente: Eficiencia en el proceso de planificación del cronograma de un proyecto de software.

Dimensión	**Indicadores**	**Unidad de Medida**
Tiempo de desarrollo de los cronogramas	Comparación con mejores tiempos	unidades de tiempo
Eficiencia en la asignación de recursos a actividades	Comparación de balance de carga entre recursos de un mismo tipo	Es correcto
		No es correcto

Métodos de Trabajo Científico

Métodos teóricos

- Analítico-Sintético: se utilizó este método en el estudio de la teoría sobre los RCPSP. Este método permitió además, arribar a conclusiones con respecto a las tendencias actuales de solución a estos problemas.
- Hipotético-Deductivo: se utilizó para corroborar la hipótesis planteada en la validación de los resultados del algoritmo de optimización que se propone.
- Modelación: Contribuye al desarrollo del algoritmo y su implementación.
- Análisis documental: permitió realizar un estudio de la bibliografía referente a la temática abordada.

Métodos empíricos

- Medición: Permite medir los resultados de la comparación el algoritmo propuesto con otros métodos existentes para analizar la eficiencia del proceso de evaluación.

- Experimental: Se utiliza con datos provenientes de casos reales. Se aplican pruebas estadísticas debidamente fundamentadas para analizar los resultados. Se establecen indicadores adecuados que permiten realizar correctas mediciones de los resultados.

Población y Muestra

El estudio de casos se realizará sobre tres instancias de prueba de la biblioteca Biblioteca de Problemas de Planificación de Proyectos (PSPLib) que contiene diferentes conjuntos de problemas para los distintos Problema de Planificación de Proyectos (por sus siglas en inglés, Project Scheduling Problem) (PSP), así como soluciones óptimas y heurísticas. Los conjuntos de datos pueden ser utilizados para la evaluación de los procedimientos de solución de los RCPSP. Fueron seleccionados 3 instancias de proyectos de 30, 60 y 120 actividades respectivamente.

Diseño de Experimentos

Para la validación de la investigación se han diseñado cinco tipos de experimentos, utilizando pre-experimentos con un grupo de experimentación y una observación después de aplicada la propuesta.

Experimento 1: Aplicación del algoritmo genético para la planificación a una instancia de proyecto de 30 actividades de la biblioteca PSPLib. El diseño se representa de la siguiente manera:

$$G \text{ ——- } X \text{ ——— } O$$

Descripción de las variables
G: Grupo de experimentación compuesto por la instancia de proyecto de la PSPLib.
X: Aplicación de la propuesta al grupo a través de la aplicación del algoritmo.
$O2$: Análisis de los resultados al aplicar la propuesta en el grupo de experimentación.

Experimento 2: Aplicación del algoritmo genético para la planificación a una instancia de proyecto de 60 actividades de la biblioteca PSPLib. El diseño se representa de la siguiente manera:

$$G \text{ ——- } X \text{ ——— } O$$

Descripción de las variables
G: Grupo de experimentación compuesto por la instancia de proyecto de la PSPLib.
X: Aplicación de la propuesta al grupo a través de la aplicación del algoritmo.
$O2$: Análisis de los resultados al aplicar la propuesta en el grupo de experimentación.

Experimento 3: Aplicación del algoritmo genético para la planificación de una instancia de proyecto de 120 actividades de la biblioteca PSPLib. El diseño se representa de la siguiente manera:

$$G \text{ ——- } X \text{ ——— } O$$

Descripción de las variables

G: Grupo de experimentación compuesto por la instancia de proyecto de la PSPLib.

X: Aplicación de la propuesta al grupo a través de la aplicación del algoritmo.

$O2$: Análisis de los resultados al aplicar la propuesta en el grupo de experimentación.

Experimento 4: Análisis del impacto económico de la propuesta. El diseño se representa de la siguiente manera:

$$G \text{ —— } X \text{ —— } O$$

Descripción de las variables

G: Grupo de experimentación compuesto por la Red de Centros de la UCI .

X: Análisis del impacto económico de la propuesta en cualquier escenario.

$O2$: Observación de la construcción de la ficha de costo de la propuesta a través del despliegue, implantación, desarrollo e integración con el GESPRO 13.05.

Experimento 5: Análisis del impacto social, lineamiento de la política económica y social del partido y la Revolución de la propuesta. El diseño se representa de la siguiente manera:

$$G \text{ —— } X \text{ —— } O$$

Descripción de las variables

G: Grupo de experimentación compuesto por la sociedad.

X: Analizar el impacto social, lineamiento de la política económica y social del partido y la revolución de la propuesta en cualquier escenario.

$O2$: Observación después del análisis social en el grupo de experimentación.

Aporte Práctico

- Esquema Generador de Soluciones en Serie (por sus siglas en inglés, Series Schedule Generator Scheme) (SSGS) desarrollado en lenguaje c++ para la generación de soluciones factibles a optimizar.

- GA desarrollado en lenguaje c++ para optimizar la planificación automática del cronograma de un proyecto de software.

Listado de publicaciones, eventos y avales de la investigación

- ESPRONCEDA, Julio C. y **LUIS, Susej Beovides**. 2014. Marco de Trabajo Ingenieril para el Desarrollo de Laboratorios Virtuales con fines educativos. Memorias de la Conferencia Científica de la Universidad de Ciencias Informáticas, Cuba. 2014, vol. 4, págs. 56-89.

- GÓMEZ, Jandy M. y **LUIS, Susej Beovides**. 2013. Desarrollo de un Laboratorio Virtual de Metalografía como herramienta de enseñanza. Memorias del Congreso Internacional Virtual de Innovación Tecnología y Educación, CIVITEC. 2013.

- **LUIS, Susej Beovides** y OJEDA, Gelson R. Saurín. 2013. Módulo de diseño en Dos Dimensiones para los Laboratorios Virtuales con fines educativos. VIII Peña Tecnológica de estudiantes y jóvenes profesionales. Cuba. 2013. Url: https://ptec.eventos.uci.cu/node/831i.

- **LUIS, Susej Beovides** y RODRÍGUEZ, Dayanis Castellanos. 2013. Procedimiento de Pruebas no funcionales para software educativo. 13ra Semana Tecnológica de FORDES Tecnologías Convergentes: Presente y Futuro. Cuba. 2013.

- OJEDA, Gelson R. Saurín; DUCUNGÉ, Ms. Orlay García; SOCARRAS, Leoder Alemañy; **LUIS, Susej Beovides** y TUGORES, Yaself Machado. 2013. Laboratorios Virtuales con fines educativos. XV Congreso de Informática en la Educación, Memorias de la XV Convención y Feria Internacional, INFORMÁTICA . Cuba. 2013.

- OJEDA, Gelson R. Saurín y **LUIS, Susej Beovides**. 2012. Módulo de localización de objetos de una escena en un mini-mapa. Desarrollo de Aplicaciones Informáticas en la Sociedad - Formación y Desarrollo, 12ma Semana Tecnológica de FORDES Tecnologías Convergentes: Presente y Futuro. Cuba. 2012. Url: http://semanatecnologica.fordes.co.cu/index.php/xiist/xii/paper/view/1015i.

- PRIETO, Dany J. Valiente; ROJAS, Onel Macías; HERNÁNDEZ, Ulises Rodríguez; ABREU, Yindra Thomas y **LUIS, Susej Beovides**. 2010. Procedimiento para la fijación de los precios de los productos y/o servicios de exportación. I Taller de Ciencias Informáticas aplicadas a la gestión empresarial, en Memorias de la V Conferencia Científica de la Universidad de las Ciencias Informáticas, UCIENCIA, Cuba. 2010.

Estructura del documento

La presente tesis está estructurada de la siguiente manera: introducción, cuatro capítulos, conclusiones, recomendaciones, bibliografía y un cuerpo de anexos, que permite abundar en los temas tratados en la misma.

El **Capítulo 1** recoge los aspectos teórico-prácticos considerados para conformar la solución que se propone al problema planteado, abordando la teoría de los Problemas de Planificación de Proyectos y los

elementos que los componen. Se hace énfasis además en el Problema de Planificación de Proyectos con Recursos Limitados.

En el **Capítulo 2** se exponen varias técnicas de optimización que han sido empleadas para dar solución a los problemas de planificación de proyectos, así como la presentación de investigaciones realizadas en este sentido, aspectos que forman la base de la solución que se propone.

En el **Capítulo 3** se describen los componentes principales de la solución propuesta para optimizar la planificación del cronograma de un proyecto de software, quedando plasmada una representación formal de las decisiones y el algoritmo de solución para el problema de optimización generado.

En el **Capítulo 4** se exponen los resultados alcanzados de la validación del algoritmo. Se valida la eficiencia y efectividad del algoritmo, a partir de la aplicación de un estudio de casos.

CAPÍTULO 1

PLANIFICACIÓN DE PROYECTOS DE SOFTWARE

1.1. Introducción

En este capítulo se recogen los aspectos teórico prácticos considerados para conformar la solución que se propone al problema planteado. Se abordan aspectos generales sobre la Planificación de Proyectos de Software, analizando el RCPSP como una variante del PSP y se explican los elementos que lo conforman y el modelo matemático de su representación.

1.2. Análisis Bibliométrico

Durante la investigación se referenciaron varios materiales. A continuación se cuantifican los materiales según la fuente consultada:

	Últimos 5 años	**Años anteriores**
Libros y monografías	13	7
Tesis de doctorados	3	2
Tesis de maestrías	5	3
Artículos en Revistas Indexadas	9	4
Memorias de eventos	7	3
Reportes técnicos y conferencias	11	2
Artículos publicados en la Web	10	4

Tabla 1.1. Tabla resumen análisis bibliométrico

Para el desarrollo de la investigación se analizaron artículos de publicaciones referenciadas, de conferencias científicas, tesis de doctorados y maestrías, libros y sitios web. Para la obtención de la documentación se realizó una búsqueda bibliográfica exhaustiva utilizando herramientas de búsqueda en internet Google

Scholar. También a través de la biblioteca de la UCI se accedió a las bases de datos de publicaciones referenciadas, tales como: Scielo, Scirus y Elsevier. Se puede apreciar abundante bibliografía referenciada contando con 83 fuentes bibliográficas consultadas, evidenciando que el 70 porciento se encuentra en el rango de fecha del 2010 hasta la actualidad. Se puede concluir que el tema ha sido abordado en investigaciones de maestrías y doctorados dentro del período de los últimos años, demostrando la pertinencia del mismo.

1.3. Planificación de Proyectos

La planeación es una actividad natural y peculiar del hombre como ser racional, que considera necesario prever el futuro y organizar su acción de acuerdo con sus previsiones, por lo que la planeación es tan antigua como el hombre mismo (GARCÍA et al., 2002).

Por medio de la planeación el hombre moderno se propone resolver problemas complejos y orientar procesos de cambio, enfrentando múltiples y complejos desafíos, haciendo un amplio uso de los recursos que le proporcionan la ciencia, la técnica y la cultura, para buscarles solución (ibíd.).

Planificar un proyecto es diseñar acciones orientadas a la consecución de determinados propósitos, procurando utilizar racionalmente los recursos disponibles. Aplicado al mundo laboral, planificar y gestionar proyectos consiste en definir objetivos productivos de corto, mediano y largo plazo en función de los cuales se programan acciones y se ordenan recursos, bajo un régimen de control de gestión y evaluación de resultados.

Por su parte, los proyectos de desarrollo de software se diferencian de los otros proyectos de ingeniería tradicional en la naturaleza lógica del producto software, pues el software se desarrolla, no se fabrica en un sentido clásico. La gestión del proyecto de software es el primer nivel del proceso de ingeniería de software, porque cubre todo el proceso de desarrollo. Para conseguir un proyecto de software fructífero se debe comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas a llevar a cabo, el esfuerzo (costo) a consumir y el plan a seguir.

La gestión de un proyecto de software comienza con un conjunto de actividades que se denominan planificación del proyecto. Antes de que el proyecto comience, se debe realizar una estimación: del trabajo a ejecutar, de los recursos necesarios y del tiempoque transcurrirá desde el comienzo hasta el final de su realización (PRESSMAN, 2005)

El objetivo de la planificación es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, coste y planeación temporal. Estas estimaciones deben definir los escenarios de mejor y peor caso de forma que los resultados del proyecto puedan limitarse. Esto permite establecer un guión temporal de las acciones y tareas a desarrollar durante la realización del software. Además, tiene en cuenta todos los factores decisivos en la ejecución de las tareas, desde su orden necesario hasta la disponibilidad de los recursos. Este calendario especifica las fechas y el orden claro de todas las tareas y actividades del proyecto.

Una correcta planificación es fundamental a la hora de optimizar los recursos en conjunto con los desarrollos y planes de trabajo. Para ello se deben tener en cuenta los correspondientes costes, limitaciones y condiciones iniciales prefijadas, y así mantener la máxima probabilidad de éxito en la consecución de los resultados y de los objetivos fijados en el proyecto (ÁLVAREZ VÁZQUEZ, 2012).

Una acertada planificación contribuye además a visualizar las futuras posibilidades y asumir cambios que favorezcan el nivel productivo del equipo; sin ella los proyectos podrían estar trabajando indefinidamente sin alcanzar los objetivos propuestos.

En cualquier caso, no debe olvidarse que la planeación tiene siempre el carácter de un medio, aunque en muchas ocasiones necesario, y no constituye un fin en sí misma. Dicho de otra manera, la planeación no se legitima por sí misma, sino en función de los acuerdos y consensos sobre los fines y objetivos que persigue. Aun logrando un consenso sobre esto, le resta todavía conseguir el acuerdo sobre los medios e instrumentos necesarios y sobre los criterios de éxito en la acción, para garantizar una evaluación adecuada de sus resultados (GARCÍA et al., 2002).

1.4. El proceso de planificación según escuelas de gestión de proyectos

Los conocimientos sobre la Gestión de Proyectos se han ido agrupando en estándares o normas, en base a recoger las buenas prácticas, reconocidas de manera generalizada, porque su aplicación de forma consistente ha demostrado su influencia en la mejora de los resultados de los proyectos.

Las escuelas de gestión de proyectos están constituidas por las normas, buenas prácticas, metodologías y estándares más representativos tanto en el plano internacional, nacional, como en la propia institución donde se desarrolla la presente investigación. En esta sección se expone un estudio comparativo de las escuelas referente a lo que plantean sobre el proceso de planificación, así como las técnicas y herramientas para su mejor realización, con especial interés en la optimización, la gestión del tiempo y la asignación de recursos, así como las herramientas informáticas que suscitan. En las siguientes subsecciones se abordan algunas descripciones y análisis de dichas escuelas.

1.4.1. Guía de los Fundamentos para la Dirección de Proyectos (PMBOK)

PMBOK (Project Management Body of Knowledge), desarrollado por Project Management Institute (PMI, 2013), proporciona pautas para el ciclo de vida del proyecto y los procesos relacionados. Constituye uno de los estándares de gestión de proyectos más reconocidos a nivel internacional. PMBOK propone controlar la ejecución del proyecto durante todo su ciclo de vida para proporcionar una retroalimentación constructiva, descubrir situaciones desconocidas o no resueltas y favorecer la mejora continua.

Los principios que propone para la planificación son coste, tiempo y ámbito. Plantea que se deben realizar definiciones previas de actividades o tareas y paquetes de trabajo o hitos. Toda planificación de proyectos debe iniciarse mediante la elaboración de una estructura de división del trabajo (WBS, por sus siglas en inglés). El WBS es un modelo visual ideal para la presentación a la dirección, dado que ofrece una

perspectiva real de todo el trabajo que el proyecto implica. Por otra parte, proporciona una estructura lógica que facilita la determinación de la duración y coste de cada tarea, además de la asignación de recursos y responsables.

Propone técnicas de optimización de recursos, costos, productos y procesos del ciclo de vida del proyecto. Recomienda reunir las lecciones aprendidas y archivar la información del proyecto para su uso futuro por parte de la organización e implementar las actividades aprobadas de mejora de procesos. PMBOK refleja además la importancia de los sistemas informáticos para mejorar o restringir las opciones de la dirección de proyectos, destacando su influencia de manera positiva o negativa sobre los resultados del proyecto. Aborda las herramientas de software para definir cronogramas, gestionar las configuraciones, recopilar y distribuir información, administrar costos y recursos, calcular indicadores de desempeño y gestionar registros de proyectos que se utilizan a lo largo del proceso de monitoreo y control del proyecto.

Si bien PMBOK registra de manera general la planificación y evaluación del desempeño del proyecto, el uso de indicadores de gestión, el trabajo con las lecciones aprendidas y la influencia de las herramientas informáticas en el éxito del proyecto; no propone mecanismos concretos ni herramientas que optimicen la planificación del proyecto dado el nivel de madurez alcanzado por la organización durante su mejora continua.

1.4.2. Modelo de Madurez y Capacidad Integrado (CMMI)

CMMI (Capability Maturity Model Integration), desarrollado por Software Engineering Institute (SEI, 2010), es un modelo para la mejora y la evaluación de procesos de desarrollo de software. Propone cinco niveles de madurez: 1-Inicial, 2-Administrado, 3-Definido, 4-Administrado cuantitativamente y 5-Optimizado. En el nivel 5 del modelo se aborda la optimización de los procesos dentro de la organización; enfocada hacia la mejora continua en el desempeño de los procesos a través de reformas innovadoras e incrementales. Plantea que los proyectos deben actualizar periódicamente la definición de sus procesos para incorporar los últimos cambios realizados por la organización a los procesos estándar cuando estos se beneficien. Sin embargo, la mejora continua debería estar presente en todo momento, no solamente cuando la organización alcanza su máximo nivel de madurez según CMMI.

El área de proceso de Planificación de Proyecto (PP) corresponde al nivel 2 en la representación por etapas y está ubicada dentro de la categoría de proceso de gestión de proyectos para la representación continua. Tiene como propósito establecer y mantener planes que definan las actividades del proyecto. Las prácticas en PP en conjunto con otras áreas son la base para la gestión del proyecto y el establecimiento de un proceso gestionado que es el fundamento para alcanzar un proceso definido.

Con PP se establecen y actualizan los planes que permiten ejecutar las actividades del proyecto. Parte de la estimación y dimensionamiento del proyecto, la generación de los planes para cubrir las diferentes actividades que se requieren y la obtención de los compromisos de los interesados respecto a los planes definidos.

Sin embargo, el modelo se enfoca en la descripción verbal de los procesos y no define métodos concretos

para el aprendizaje y la adaptación durante el control de la ejecución de proyectos según el nivel de madurez alcanzado por la organización. Expone la importancia de las herramientas informáticas como apoyo a la toma de decisiones del equipo de desarrollo del proyecto para administrar el proyecto, diseñar, gestionar requisitos, evaluar y probar. Refleja además la importancia de la mejora tecnológica para dar soporte a la optimización de los procesos como una gestión sistemática de la organización. Aborda los sistemas de información como herramienta para acelerar y compartir el aprendizaje, sobre todo en procesos ágiles e innovadores donde se depende de la participación de empleados que estén adaptados al negocio y objetivos de la organización. Sin embargo, deja a las organizaciones la responsabilidad de implementarlas.

1.4.3. Proyectos en Entornos Controlados (PRINCE2)

La metodología PRINCE2 (Projects in Controlled Environments), establece un conjunto de buenas prácticas en torno a la gestión de proyectos, que cubre el control, la administración y la organización de proyectos (PRINCE2, 2009).

PRINCE2 está basado en los productos, es decir, los planes del proyecto se centran en obtener resultados concretos, y no sólo en la planificación de las actividades que se llevan a cabo. La planificación es un documento, enmarcado en un método o esquema predefinido, que describe cómo, cuándo y quién es responsable de conseguir una serie de metas.

PRINCE2 describe tres etapas en la técnica de planificación:

- Producir una estructura de desglose del producto (PBS): que no es más que realizar una jerarquía de los productos que el plan tiene que producir.
- Describir el producto: que conlleva escribir para cada producto identificado, una descripción para todos los niveles de la estructura de desglose.
- Producir un diagrama de flujo del producto (PFD): que implica elaborar un diagrama que muestra la secuencia por la que los productos tienen que ser desarrollados y las dependencias entre ellos.

PRINCE2 es una metodología en la que se puede tener un inicio, un desarrollo y un cierre de proyecto controlable, así como una forma de poder revisar el avance del mismo, midiéndose contra los planes del proyecto y la justificación del mismo. También se pueden tomar decisiones en el momento más adecuado y tener bajo control cualquier desviación versus los planes originales así como mantener informado adecuadamente a los directivos del proyecto con una comunicación efectiva. Pero no provee de herramientas como lo son el uso de diagramas de Gantt, análisis de redes, análisis financiero, cuadros de riesgo, etc. Más bien esta metodología deja abierto para que cada gerente de proyecto utilice las herramientas que desee, ya que de igual forma las utilizará para el desarrollo del mismo, pero no limita su uso.

1.4.4. Organización Internacional de Normalización (ISO)

La norma ISO 10006:2003 desarrollada por la Organización Internacional de Normalización (International Standards Organization, ISO por sus siglas en inglés) proporciona orientación para gestionar la calidad durante el desarrollo de los proyectos (ISO, 2003).

Plantea que la planificación es el procedimiento para evaluar las necesidades del cliente, establecer la planificación de trabajos y poner en marcha los restantes procedimientos. Además va a gestionar la comunicación y los conflictos entre los participantes, medir y evaluar el desarrollo del proceso proyecto-construcción y tomar medidas para canalizar las desviaciones. Se encargará además de identificar, documentar y aprobar la necesidad de llevar a cabo modificaciones en el proceso y revisar su implementación, así como asegurarse de que los procedimientos finalicen cuando se prevé y que la documentación se ha guardado y almacenado convenientemente. Propone también que se documenten e incluyan en el plan de gestión de proyectos los planes de recursos, incluyendo la estimación, las asignaciones y las limitaciones, junto con las suposiciones de que se parte.

La norma reconoce la importancia de los sistemas informáticos como apoyo a la toma de decisiones. En orden de mejorar el rendimiento de la organización del proyecto, plantea que se debe proveer al personal, de las herramientas, técnicas y métodos apropiados para facilitar el monitoreo y control de los procesos.

ISO 10006 es un estándar que busca la calidad mucho antes de finalizar el producto, ya que los procesos para producir el producto con la calidad necesaria están estandarizados de tal manera que se asegure la calidad del mismo. También se denotan las ventajas que proporciona, ya que los procesos están reglamentados, pero no detalla de una manera eficaz cada proceso efectuado, lo cual podría traer consecuencias negativas. Su éxito radica, en la manera en que se aplica y la experiencia que se tenga con otras normas ISO o estándares de calidad.

Por su parte la norma ISO 21500:2013 establece una guía para la dirección y gestión de proyectos; describe conceptos y procesos que forman parte de las buenas prácticas (STELLINGWERF y ZANDHUIS, 2013). Identifica los procesos de dirección y gestión (Inicio, Planificación, Implementación, Control, Cierre) agrupados por grupos de materia (Integración, Parte interesada, Alcance, Recurso, Tiempo, Costo, Riesgo, Calidad, Adquisiciones, Comunicación) que permiten distribuir y gestionar la información relevante del proyecto estableciendo entradas y salidas. Puede ser usada como acoplamiento entre diferentes procesos de negocio y gestión de proyectos.

Plantea que las prácticas y métodos de gestión de proyectos desarrollados por la organización requieren ser mejorados en dependencia de la experiencia alcanzada, sin que implique realizar cambios drásticos. Enfatiza el rol que juegan las condiciones del entorno durante la ejecución de proyectos en orden de maximizar el valor agregado de sus entregables. En los métodos y prácticas de gestión propone ajustar solo lo necesario para el proyecto y entorno específico, documentando los cambios o adiciones transparentemente.

La norma ISO 21500 provee orientaciones generales sobre la disciplina de la administración de proyectos. Define procesos así como entradas y salidas, más no herramientas ni técnicas.

Sin embargo, al igual que sucede con las escuelas analizadas hasta este momento, no propone meca-

nismos concretos que aseguren una acertada planificación de proyectos, así como tampoco la asignación de recursos durante la planeación de los mismos.

1.4.5. Normas Cubanas para la gestión de proyectos (NC)

Las normas cubanas para la gestión de proyectos están determinadas por un conjunto de resoluciones y decretos aprobados por ministerios. La especificación de dichas normas se orienta al tipo de proyecto que desarrolla el ministerio en cuestión. En el caso de los proyectos de desarrollo de software le compete al Ministerio de la Informática y las Comunicaciones (MIC) la determinación de las normas a seguir. Para dar a conocer la actualización de las normas cubanas se hace público el boletín NCOnline (Normas Cubanas Online) en (CUBAINDUSTRIA, 2014).

Hasta enero de 2015 existen registradas 4527 normas vigentes en Cuba, de las cuales 310 se encuentran relacionadas con la gestión de proyectos de diversas ramas de la industria, siendo NC ISO 10006:2007 la norma más actual. Se destaca el reconocimiento de las normas ISO como la opción más viable a seguir para organizar los procesos de la producción. Esto se debe fundamentalmente a que ISO ofrece guías y permite obtener certificaciones y evaluaciones que ayudan a mejorar la imagen de la empresa ante el mercado nacional e internacional. El Decreto No. 327 del año 2014 (MINISTROS, 2014) constituye el Reglamento del proceso inversionista en Cuba, teniendo como objeto contribuir a su eficiencia, racionalidad e integralidad.

El decreto plantea que todos los sujetos del proceso inversionista deben poseer preparación en las técnicas de evaluación, selección y gestión de la tecnología que garantice el enfoque integral de la inversión y la organización eficiente del proceso inversionista. Sin embargo, al igual que sucede con la mayoría de los estándares internacionales, las normas cubanas relacionadas con la gestión de proyectos representan guías abstractas que permiten ordenar y perfeccionar el trabajo, pero no ofrecen soluciones concretas. Esto hace que surjan brechas entre la práctica social y el marco normativo existente. Por otra parte, dichas normas no proponen mecanismos ni herramientas que aseguren la adaptación en el control de la ejecución de proyectos dado el nivel de madurez alcanzado por la organización durante su mejora continua.

Las actuales políticas socio-económicas de Cuba hacen un llamado a perfeccionar los trabajos de planificación, erradicando la espontaneidad, la improvisación, la superficialidad y el incumplimiento de dichos planes. De igual manera, demandan el fortalecimiento de los procesos de control sistemáticos y el incremento de la soberanía tecnológica (PCC, 2011).

1.4.6. Programa de Mejora de la UCI

El Programa de Mejora en la UCI comienza en el 2008 con el objetivo de alcanzar la certificación internacional del nivel 2 de CMMI. Esta meta se logra en el 2011 y convierte a la UCI en la primera empresa cubana certificada con este modelo y una de las pocas en el área del Caribe. Para ello se necesitaron definir procedimientos que permitieran dar cumplimiento a los requisitos que establece esta área de proceso (RAMOS, 2011).

El desarrollo de software dentro de la actividad productiva de la UCI se caracteriza por el uso de diferentes metodologías de desarrollo entre robustas y ágiles. A pesar de la variedad de metodologías usadas, se ha comprobado que muy pocos proyectos la aplican en su totalidad. Es por ello que se decide definir una metodología de desarrollo única para la actividad productiva de la UCI, que responde a una variación del Proceso Unificado Ágil (Agile Unified Process, AUP) (EDEKI, 2013) y apoyado en el modelo CMMI v1.3 (SEI, 2010).

Como parte del Programa de Mejora de Procesos, basado en CMMI-DEV, llevado a cabo, se implantaron los procesos Planeación de Proyecto (PP) y Monitoreo y Control de Proyecto (PMC). El proceso de PP consta de 7 subprocesos que garantizan el cumplimiento de cada práctica específica del área según el nivel 2 de CMMI, mediante la realización de varias actividades. Los procesos de PP y PMC incluyen guías que explican cómo: elaborar el cronograma, monitorear y controlar el proyecto, realizar reportes y administrar las acciones correctivas adecuadamente. Los mismos proponen roles y sus funciones para realizar cada actividad. Pero los libros de procesos como las guías omiten la explicación de cómo planificar y controlar el proyecto haciendo uso de una herramienta informática de manera general, dificultando el uso adecuado del conocimiento y la tecnología como apoyo al mismo.

Del análisis de las principales escuelas en la gestión de proyectos se puede concluir que todas de una forma u otra hacen alusión a la planificación dentro de la gestión de proyectos. Todas registran de manera general la realización de planes, la gestión del tiempo y de los recursos, la importancia de la mejora continua, el trabajo con lecciones aprendidas y la influencia de las herramientas informáticas en el éxito de los proyectos como apoyo a la toma de decisiones. Pero si bien PMBOK y PRINCE2 proporcionan buenas prácticas para la gestión de proyectos de manera general, no especifican acciones concretas a tener en cuenta en el desarrollo de proyectos de software. Por su parte, pese a la gran ventaja que CMMI y las ISO proporcionan en condición de valor agregado a las entidades, su aplicación resulta costosa en términos económicos y de esfuerzo, requiriendo gran inversión de capital, tiempo y recursos y el retorno de la inversión se produce a largo plazo. Se puede apreciar además, que solo PRINCE2 propone roles concretos, PMBOK es el único que describe indicadores para controlar el proyecto y ningún modelo o estándar de los analizados proponen y/o especifican herramientas informáticas como apoyo al proceso de planificación y control de proyectos.

1.5. Problema de Planificación de Proyectos

Entre las técnicas de modelado de proyectos más empleadas en la actualidad se encuentra el problema de programación de tareas o scheduling, llamado en la comunidad científica como PSP. El PSP puede definirse de manera muy general como el problema de organizar o secuenciar una serie de operaciones y ubicarlas en el tiempo sin violar ninguna de las restricciones, de precedencias y recursos, impuestas en el sistema. Dicho problema está formado por actividades, recursos, relaciones de precedencia y una función objetivo.

Las primeras aproximaciones fueron desarrolladas en los años 1950. Algunos clásicos son (KELLEY JR, 1961; MALCOLM; ROSEBOOM; CLARK y FAZAR, 1959; MODER y PHILLIPS, 1964; PRITSKER;

WAITERS y WOLFE, 1969). Los PSP constituyen una familia de diversos problemas según su objetivo, los tipos de recursos disponibles y la topología de la red. Esta familia de problemas presenta distintas variantes, que van desde la simple planificación de actividades sin tener en cuenta los recursos que consumen, hasta variantes más sofisticadas que consideran varios modos de procesamiento de las actividades, generalización de las relaciones de precedencia, múltiples proyectos de forma simultánea y proyectos con recursos variables (S. HARTMANN y BRISKORN, 2010).

Al crear un modelo que represente las decisiones tácticas y operativas en la planificación de proyectos se hace indispensable abordar sobre el formalismo que permita modelar: cómo establecer la secuencia de las actividades y cómo asignar los recursos en los proyectos que se ejecutan simultáneamente. Para este propósito se emplea lo que establece la teoría de los Problemas de Planificación de Proyectos (MEDRANO BROCHE, 2012).

1.6. Elementos de los problemas de planificación de proyectos

1.6.1. Actividades

Los proyectos cuentan con un número finito de actividades (tareas, trabajos u operaciones) que se necesitan ejecutar (CAPETILLO CORBEA, 2012). Para terminar un proyecto es necesario ejecutar cada actividad. También, las actividades determinan los requerimientos de recursos para su ejecución. Lo que se debe hallar en un PSP es cuándo y de qué modo se va a procesar cada actividad.

1.6.2. Relaciones de precedencia

En los proyectos al intentar planificar una actividad, la misma puede o no estar condicionada a la ejecución de otra actividad. Esta relación entre las actividades se denomina relación de precedencia. La forma más simple en que pueden aparecer las relaciones de precedencia es cuando una actividad no puede comenzar hasta que otra(s) finalice(n) (DEMEULEMEESTER, 2002).

Por ejemplo la actividad j solo se puede ejecutar a partir de que finalice la actividad i, este tipo de relación es de *fin-inicio* (el final de la actividad i restringe el inicio de la actividad j). Si la actividad j comienza inmediatamente después de la terminación de i, se dice que el lapso de tiempo entre actividades es cero. Existen relaciones de precedencia que permiten mayor flexibilidad y son denominadas relaciones de precedencia generalizadas (BLAZEWICZ; ECKER; H.; G. y WEGLARZ, 2007). En este trabajo se consideran las relaciones de precedencia del tipo *fin-inicio*, con lapso de tiempo entre las actividades igual a cero.

Las relaciones de precedencia se representan mediante un grafo dirigido y sin ciclos, en formato Actividad En Nodo (por sus siglas en inglés, Activity On Node) (AON) (MALCOLM; ROSEBOOM; CLARK y FAZAR, 1959). Cada actividad representa un nodo del grafo y cada arco dirigido una relación de precedencia entre dos actividades. Cada arco tiene asociado el tipo de relación de precedencia que existe entre las dos actividades que separa.

1.6.3. Recursos

Para realizar una actividad, generalmente es necesario tener algún tipo de consumo. En los PSP se modela ese consumo a través de los recursos que utiliza cada actividad. Los recursos se clasifican según su categoría, tipo y valor (KOLISCH y PADMAN, 2001). Las categorías más generales de los recursos son las de recursos renovables y no renovables.

Recursos renovables:

La disponibilidad de estos recursos está limitada en cada unidad de tiempo. Sin considerar la duración del proyecto, como restricción cada recurso renovable está disponible en una cierta cantidad constante en cada unidad de tiempo y su utilización no puede exceder esa cantidad en ninguna de esas unidades (MEDRANO BROCHE, 2012). Como ejemplo podemos citar: mano de obra, máquinas, herramientas. En esta investigación solo se tendrán en cuenta los recursos que responden a esta clasificación.

Recursos no renovables

Recursos que están limitados a lo largo del ciclo de vida del proyecto, y una vez usados en el procesamiento de una actividad no pueden ser asignados a otra (FRANCISCO, 2002). Ejemplo de ellos lo constituyen: presupuesto, materia prima, energía.

1.6.4. Funciones objetivos

Todos los problemas de optimización consisten en encontrar una mejor solución respecto a un criterio determinado. Una función objetivo cuantifica la calidad de una solución, el criterio por el cual se mide su bondad. Según las clasificaciones dadas en (BRUCKER; DREXL; MÖHRING; NEUMANN y PESCH, 1999), cada función objetivo define un modelo distinto, aunque el conjunto de soluciones posibles sea el mismo. Diferentes funciones objetivos pueden requerir técnicas distintas para resolver el problema (FRANCISCO, 2002).

La minimización de la duración del proyecto (tiempo transcurrido entre el inicio y la terminación del proyecto) es probablemente la función objetivo más tratada y aplicada en el dominio de la planificación de proyectos (HARTMANN y BRISKORN, 2008; KOLISCH y PADMAN, 2001). La duración del proyecto se define como el tiempo que ha de transcurrir entre la ejecución de la primera y última tarea. Como el inicio del proyecto se sitúa usualmente en $t = 0$, minimizar la duración del proyecto equivale a minimizar el máximo de los tiempos finales de las actividades. Las funciones objetivos que dependen del tiempo en los PSP, son clasificadas como *regulares*[1] (MEDRANO BROCHE, 2012). En esta investigación se empleará siempre una función objetivo regular.

[1] *Una función objetivo regular, es aquella que al comparar dos secuencias S y $\hat{S}$ para un problema dado, si S difiere de $\hat{S}$ únicamente en el tiempo de finalización de una actividad j y se cumple que $f_j < \hat{f}_j$ para $f_j \in S$ y $\hat{f}_j \in \hat{S}$, se puede asegurar que la secuencia con el menor tiempo de finalización en esa actividad es al menos tan buena como la otra secuencia (tomando la evaluación de la función objetivo como criterio) y se dice que S domina a $\hat{S}$*

1.7. Problema de Planificación de Proyectos con Recursos Limitados

El problema PSP, ha sido ampliamente estudiado en la literatura, es un problema combinatorio clásico que aparece en las líneas de manufactura y producción, en particular en líneas de ensamble o sistemas de producción Justo a Tiempo (por sus siglas en inglés, Just In Time (JIT).La formulación de modelos matemáticos ha sido abordada por diversos autores (POSADA, 2010).

Dentro de los problemas de planificación de actividades, uno de los más estudiados es el de planificación de tareas con recursos restringidos (FRANCISCO, 2002). El problema básico consiste en minimizar la duración del proyecto cuando las actividades que lo componen no pueden interrumpir su ejecución y están sujetas exclusivamente a relaciones de precedencia de tipo Fin-Inicio. Las actividades utilizan para su ejecución un conjunto de recursos renovables con disponibilidad limitada y constante a lo largo de todo el proyecto. Las actividades tienen un único modo de ejecución, con una duración determinada y un consumo dado de recursos. También se suele hacer referencia a él como la versión estándar del problema RCPSP o la versión de modo único (POSADA, 2010).

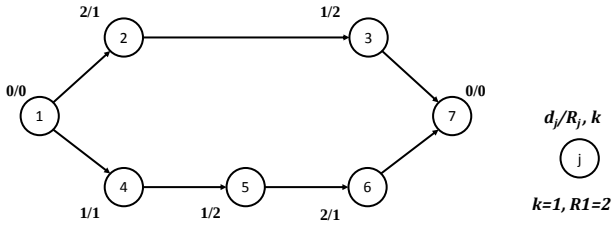
El RCPSP consiste en establecer la secuencia de un conjunto de actividades $J=\{1,...,n\}$ de un proyecto, sujetas a dos tipos de restricciones, las relaciones de precedencia y la cantidad de recurso disponible para ejecutar las actividades en cada instante de tiempo y se asumen que los recursos son renovables de diferentes tipos. Como función objetivo se tiene la minimización del tiempo de terminación del proyecto. Todas las magnitudes son asumidas enteras.

1.7.1. Representación del RCPSP

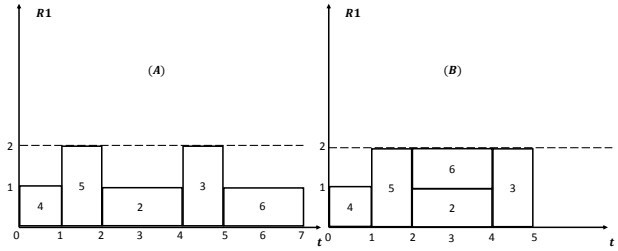
Un proyecto es representado por un grafo dirigido sin ciclo $G = (J, E)$ donde J es el conjunto de los nodos (actividades) y $E = \left\{(i, j) : i \in A_j\right\}$ el conjunto de los arcos (relaciones de precedencia), donde $A_j \subset J$ es el conjunto de las actividades antecesoras de j y por ende tienen que ejecutarse antes que esta. El conjunto de las actividades sucesoras inmediatas de j denotado por Suc_j cada actividad j tiene una duración d_j (peso del arco (i, j)) con $d_0 = d_n = 0$ ya que $j = 1$ y $j = n$ son nodos ficticios que son tomados para representar donde comienza y termina el proyecto. Se denota como R_k al conjunto de recursos disponibles del tipo $k \in K$. Cada actividad j para procesarse consume una cantidad q_{kj} de recurso. A lo largo de este trabajo denotaremos por s_j $\left(f_j\right)$ al tiempo de inicio (fin) de la actividad j en una secuencia $(s_1, s_2, ..., s_n)$, $(f_1, f_2, ..., f_n)$ (MEDRANO BROCHE, 2012).

En 1.1a se representa la red de un proyecto con siete actividades y dos recursos del mismo tipo disponibles y que denotaremos como $(P1)$. En 1.1b se muestran dos diagramas de Gantt para dos soluciones.

Para el caso del proyecto $P1$ las soluciones que aparecen representadas por los diagramas de Gantt están dadas por los vectores de tiempos de inicio $S_a = (0, 2, 4, 0, 1, 5, 5)$, $S_b = (0, 2, 4, 0, 1, 2, 4)$ respectivamente. Note que los tiempos de inicio y de fin de las actividades ficticias $j = 1$ y $j = 7$ coinciden con el inicio y fin de la secuencia.



(a) Red de proyecto (P1)



(b) Solución (P1)

Figura 1.1. Red y diagrama de Gantt de un proyecto (tomado de (BALLESTÍN GONZÁLEZ, 2002))

1.7.2. Modelo matemático del RCPSP

Sea $P_t = \left\{ j \in J : \ s_j \leq t < s_j + d_j \right\}$ el conjunto de las actividades que se están procesando en el instante de tiempo t y $U(S,t) = \sum_{j \in P} q_{jk}$ la cantidad de recurso que consumen estas. El RCPSP se puede definir de la siguiente manera:

$$min \ s_n \tag{1.7.1}$$

Sujeto a:

$$s_j - s_h \geq d_j \ \ \forall h \in A_j, j \in J \tag{1.7.2}$$

$$U(S,t) \leq |R_k|, \ t = 0, 1, 2, ..., T-1, \forall k \in K \tag{1.7.3}$$

$$s_j \geq 0 \ \ \forall j \in J \tag{1.7.4}$$

La expresión 1.7.1 define la minimización de la duración del proyecto, que está dada por el tiempo de inicio de la actividad ficticia que pone fin a la red del proyecto. El grupo de restricciones 1.7.2 controlan las relaciones de precedencia en la secuencia y las restricciones 1.7.3 controlan que no se exceda la cantidad de recurso disponible para cada instante t. Finalmente 1.7.4 define la no negatividad de los tiempos de inicio (FRANCISCO, 2002).

1.8. Conclusiones del capítulo

- En este trabajo se seguirán los fundamentos propuestos por la teoría de planificación de proyectos.
- Del análisis de las principales escuelas en la gestión de proyectos se puede concluir que todas de una forma u otra hacen alusión a la planificación dentro de la gestión de proyectos, pero ningún modelo o estándar de los analizados proponen y/o especifican herramientas informáticas como apoyo al proceso de planificación y control de proyectos.
- A partir del análisis de los principales elementos de la planificación de proyectos de software se concluye que el RCPSP es el problema a emplear para modelar la problematica existente.

CAPÍTULO 2

METAHEURÍSTICAS PARA PROBLEMAS DE PLANIFICACIÓN

2.1. Introducción

En este capítulo se recogen los aspectos teórico-prácticos considerados para conformar la solución que se propone al problema planteado. Se abordan aspectos generales sobre la Planificación de Proyectos de Software, analizando el RCPSP como una variante del PSP y se explican los elementos que lo conforman y el modelo matemático de su representación. Se describen métodos de optimización para la solución de dicho problema, basados en metaheurísticas, así como la presentación de investigaciones realizadas en este sentido.

2.2. Metaheurísticas para la solución de RCPSP

Los problemas de planificación de proyectos con recursos limitados, han sido estudiados en la literatura y existen numerosos trabajos que lo abordan desde diferentes perspectivas. Los enfoques que se han usado para su solución son exactos y de aproximación. El primero garantiza una solución óptima, siempre que esta exista; sin embargo, en problemas grandes o muy complejos podría hacerse inviable ya que el tiempo de cómputo crece en forma exponencial con el tamaño del problema. El segundo se traduce en metodologías heurísticas o metaheurísticas que si bien no garantizan la optimalidad, pueden dar muy buenas soluciones en tiempos razonables (MORILLO; MORENO y DÍAZ, 2014).

El término metaheurísticos fue utilizado por primera vez por Glover (GLOVER, 1986) y su significado ha cambiado a lo largo de los años. En la actualidad, un algoritmo metaheurístico puede ser visto como una estrategia inteligente para diseñar o mejorar procedimientos heurísticos con un alto desempeño (MELIÁN; PÉREZ y VEGA, 2003). Por lo general combinan métodos constructivos, métodos de búsqueda local, conceptos que vienen de la Inteligencia Artificial, Métodos Estadísticos y Métodos Bioinspirados.

Los algoritmos metaheurísticos son procedimientos en los cuales se introducen cierta inteligencia que permite tanto la exploración, para la búsqueda en un gran espacio de soluciones factibles, como la explotación, para concentrar la búsqueda en una región reducida del espacio factible donde puede estar la solución

óptima (MORILLO; MORENO y DÍAZ, 2014). Además, integran mecanismos para escapar de óptimos locales. Estos algoritmos son los más conocidos en la literatura para la solución de múltiples problemas complejos, como los combinatorios, grupo al cual pertenece el RCPSP.

Algunas de las razones de la popularidad y éxito de estos algoritmos metaheurísticos son su gran versatilidad, que permite su implementación para la solución de diversos tipos de problemas, la relativa simplicidad de sus conceptos subyacentes y un amplio espectro de variantes en sus aplicaciones (SANTANA; RODRÍGUEZ; LÓPEZ et al., 2004).

2.2.1. Clasificación de las metaheurísticas

Hay diferentes formas de clasificar y describir las técnicas metaheurísticas. Dependiendo de las características que se seleccionen se pueden obtener diferentes taxonomías: basadas en la naturaleza y no basadas en la naturaleza, con memoria o sin ella, con una o varias estructuras de vecindario, etc. Una de las clasificaciones más populares las divide en metaheurísticas basadas en trayectoria y basadas en población. Las primeras manipulan en cada paso un único elemento del espacio de búsqueda, mientras que las segundas trabajan sobre un conjunto de ellos (población). Esta taxonomía se muestra de forma gráfica en 2.1, que además incluye las principales metaheurísticas (J. F. CHICANO, 2007).

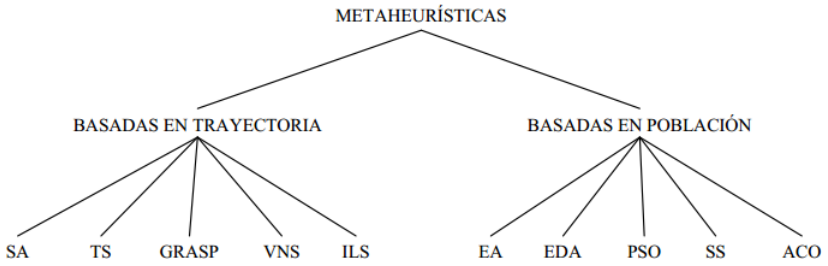


Figura 2.1. Clasificación de las técnicas de Optimización (J. F. CHICANO, 2007)

En este trabajo se hace hincapié en las metaheurísticas más relevantes que se encuentran en la literatura para la solución del RCPSP tanto de las basadas en trayectoria como en población.

2.3. Metaheurísticas basadas en trayectoria

La principal característica de estos métodos es que parten de una solución y, mediante la exploración del vecindario, van actualizando la solución actual, formando una trayectoria. La mayoría de estos algoritmos surgen como extensiones de los métodos de búsqueda local simples a los que se les añade algún mecanismo para escapar de los mínimos locales. Esto implica la necesidad de una condición de parada más elaborada que la de encontrar un mínimo local. Normalmente se termina la búsqueda cuando se alcanza un número

máximo predefinido de iteraciones, se encuentra una solución con una calidad aceptable, o se detecta un estancamiento del proceso.

2.3.1. Recocido Simulado (Simulated Annealing)

El algoritmo Recocido Simulado fue introducido por (KIRKPATRICK; GELATT y VECCHI, 1983). Se inspira en el proceso de templado del metal, específicamente en el comportamiento molecular de un metal fundido cuando es sometido a un enfriamiento súbito. El algoritmo empieza desde una solución inicial aleatoria x_0. A partir de esta, se define una vecindad de soluciones a su alrededor y se evalúa x_0 en la función objetivo $f(x_0)$. Se genera una nueva solución x_1 perteneciente a la vecindad de x_0 y se evalúa la nueva solución $f(x_1)$. Si $f(x_1)$ es mejor que $f(x_0)$, entonces se acepta la nueva solución y se continúa el proceso a partir x_1. De lo contrario, la nueva solución se acepta con una cierta probabilidad, la cual depende de la diferencia de las funciones objetivo, $f(x_0)$ y $f(x_1)$, y de un parámetro T llamado temperatura, la cual debe definirse al inicio del algoritmo y que varía adaptativamente durante todo el proceso. Este procedimiento se repite para n iteraciones o hasta que se cumpla un criterio de parada (MORILLO; MORENO y DÍAZ, 2014).

En la expresión 2.3.1 se representa el esquema general de la probabilidad de aceptar una solución peor, en el caso de un problema de minimización, donde n es el número de iteraciones y $f(x_n)$ es la función objetivo evaluada en la solución x_n.

$$P(\text{aceptar una solución peor}) = e^{\frac{f(x_{n-1})-f(x_n)}{T}} \tag{2.3.1}$$

La expresión 2.3.1 muestra que mientras mayor sea la diferencia entre $f(x_{n-1})$ y $f(x_n)$ menor será la probabilidad de aceptación de una nueva solución que no mejore la actual. Para un problema de maximización, el numerador del exponente en la expresión 2.3.1 deberá escribirse como $f(x_n) - f(x_{n-1})$.

La temperatura, por lo general, empieza con un valor grande, lo que permite aceptar con mayor probabilidad soluciones que no mejoran, y con ello explorar el espacio factible en las primeras etapas del algoritmo, evitando estancamientos en óptimos locales. Esta temperatura se reduce de manera gradual a medida que evoluciona el algoritmo, mediante un parámetro r, menor que 1, cuyo valor representa la velocidad de enfriamiento del algoritmo, cambiando su perfil de exploración a uno de explotación. Este procedimiento se repite hasta que se cumpla un criterio de parada. Generalmente, en el Recocido Simulado se define una temperatura mínima, denominada temperatura de congelación, a la cual deberá parar el algoritmo cuando ésta se alcance por primera vez. En (BROOKS y MORGAN, 1995; KIRKPATRICK; GELATT y VECCHI, 1983) se exponen los resultados de emplear un algoritmo Recocido Simulado en la solución de problemas RCPSP.

2.3.2. Búsqueda Tabú (Tabu Search)

El desarrollo de esta metaheurística es atribuido a Glover en (GLOVER, 1989), es una extensión de la búsqueda local que trata de no quedar atrapado en un óptimo local, mediante el uso de información que recauda a medida que el algoritmo avanza. Por esta razón, la búsqueda tabú es uno de los enfoques más representativos de los algoritmos con memoria (MORILLO; MORENO y DÍAZ, 2014).

El algoritmo empieza con una solución inicial aleatoria x_0. Posteriormente, se define una vecindad inicial de soluciones alrededor de x_0 y una lista tabú. Luego, mediante reglas de movimiento, se generan nuevas soluciones las cuales se analizan para determinar si se seleccionan. La lista tabú se construye con las soluciones históricamente visitadas, las cuales quedan prohibidas (definidas como tabú), de manera que restringen el vecindario actual; esto garantiza, hasta cierto punto, no volver a seleccionar aquellas soluciones visitadas antes, y así evitar quedar atrapado en un óptimo local. Esta lista tabú es de tamaño limitado, cuando éste se alcanza, se elimina la solución más antigua (que deja de ser tabú) y se adiciona la nueva.

Para aumentar la eficiencia en la etapa de explotación, generalmente, en la lista tabú no se almacenan directamente las soluciones, sino sus características inherentes. Así, cualquier solución que comparta algunas características que estén en la lista tabú no se selecciona. Esta regla solo tiene una excepción: cuando la nueva solución supera la mejor solución encontrada hasta el momento, se le asigna una probabilidad de ser seleccionada, denominada nivel de aspiración (ibíd.).

En (ARTIGUES; MICHELON y REUSSER, 2003) se usan técnicas de inserción para desarrollar un procedimiento de búsqueda tabú en la solución del RCPSP que, de manera iterativa, selecciona una actividad de una solución y la elimina de la programación; luego, la reinserta mediante el uso de reglas de flujo de redes. Tanto la actividad seleccionada como sus predecesores y sucesores se adicionan a la lista tabú. En (BAAR; BRUCKER y KNUST, 1999) se diseñan dos esquemas de búsqueda tabú para resolver el RCPSP; el primero genera vecindad con ayuda de la información de la ruta crítica, y el segundo se basa en la generación de vecinos. Mientras que (KLEIN, 2000) propone la llamada Búsqueda Tabú Reactiva (Reactive Tabu Search), basada en la representación serial de actividades mediante un Esquema Generador de Soluciones (por sus siglas en inglés, Schedule Generator Scheme) (SGS). Las vecindades de soluciones se obtienen con movimientos de intercambio que incluyen desplazamientos de los predecesores y sucesores de las actividades intercambiadas.

El uso de los algoritmos de búsqueda tabú en el RCPSP ha dado resultados interesantes, los que obtienen mejores resultados en la librería PSPLib utilizan la representación de lista de actividades y el esquema serie para decodificar la solución (**cervantes2010nuevos**).

2.3.3. Procedimientos de Búsqueda Voraces Aleatorizados y Adaptativos (GRASP)

GRASP proviene de las siglas de Greedy Randomized Adaptive Search Procedures. Los GRASP fueron descritos en (FEO y RESENDE, 1989). Es útil en problemas que pueden descomponerse en una secuencia de decisiones que deben tomarse en varias etapas. Este método consta de dos fases fundamentales, una fase constructiva y una de mejora (MORILLO; MORENO y DÍAZ, 2014).

En la fase constructiva se generan soluciones factibles de manera iterativa, donde en cada iteración se incorpora un elemento a la solución, cuya selección se basa en el valor de la función de aptitud parcial de cada alternativa disponible. En lugar de seleccionar directamente la mejor alternativa (metodología avariciosa, Greedy) selecciona de manera aleatoria el siguiente elemento a incorporar como parte de la solución, pero asigna una probabilidad en proporción a la función de aptitud. Una vez se elige el elemento a adicionar en la solución, los valores de las alternativas se adaptan a la nueva situación.

Una vez construida la solución, se le aplica una búsqueda local o una función de mejora, estos dos pasos se repiten un número de veces predeterminado. Al emplear componentes aleatorios, en general se obtendrán diferentes soluciones en cada iteración (FRANCISCO, 2002).

2.4. Metaheurísticas basadas en población

Los métodos basados en población se caracterizan por trabajar con un conjunto de soluciones (denominado población) en cada iteración, a diferencia de los métodos basados en trayectoria, que únicamente manipulan una solución del espacio de búsqueda por iteración.

2.4.1. Búsqueda Dispersa (Scatter Search)

La Búsqueda Dispersa fue propuesta en (GLOVER, 1998) y descrita en (LAGUNA; MARTI y MARTÍ, 2003). Esta metaheurística se considera un proceso evolutivo donde se construye un conjunto de referencia de soluciones buenas, pero dispersas, es decir, soluciones distanciadas entre sí o significativamente diferentes. Este conjunto evoluciona combinando dos o más soluciones para producir otras mejores, pero mantiene un significativo grado de dispersión (MORILLO; MORENO y DÍAZ, 2014).

Según la descripción que se ofrece en (FRANCISCO, 2002), la búsqueda dispersa consta de 5 elementos:

- Un método de generación diversificada para generar una colección de soluciones de prueba diversas.
- Un método de mejora que transforma una solución de prueba en una o más soluciones mejoradas.
- Un método para construir, mantener y actualizar el conjunto conformado por un número pequeño de las mejores soluciones encontradas.
- Un método que, a partir del conjunto de referencia, genera subconjuntos con el fin de combinarlos para crear nuevas soluciones.
- Un método de combinación de soluciones que transforma un subconjunto dado de éstas en una o más soluciones. Este método de combinación es el equivalente al operador genético de cruce en los GA.

Según (DEBELS; DE REYCK; LEUS y VANHOUCKE, 2006), este algoritmo se ha implementado para resolver el RCPSP con buenos resultados, situándose en la literatura como un método tan competitivo como los GA para este tipo de problemas.

2.4.2. Algoritmos Genéticos (GA)

Debido a las dificultades computacionales de las soluciones exactas al problema de planificación de proyectos con recursos limitados, y a que no se ha encontrado un algoritmo heurístico general, que funcione bien para todo tipo de proyectos, sigue habiendo interés en desarrollar metaheurísticos como los algorítmos genéticos (J. HOLLAND, 1975), que buscan la mejor solución en cada caso.

Los GA introducidos por Holland (J. H. HOLLAND, 1992), proporcionan un mecanismo de búsqueda robusto, para resolver problemas de optimización. Se inspiran en la teoría darwiniana de la evolución, trasladada al campo de optimización. Se puede interpretar así: cada individuo (solución) tiene asociado un valor de su aptitud, representado por su valor en la función objetivo. Las diferentes generaciones de individuos evolucionan mediante la selección y operadores genéticos tomando varias soluciones simultáneamente, logrando un proceso de búsqueda en paralelo (MORILLO; MORENO y DÍAZ, 2014).

La implementación de estos algoritmos se basa en la representación de un conjunto de soluciones en un código genético, expresado en lenguaje computacional. En la literatura se conocen dos formas de codificar las soluciones, la representación binaria, propuesta en (J. HOLLAND, 1975), y la representación en valor real. La diferencia de estos dos métodos, consiste en la manera de definir los denominados operadores genéticos.

En este algoritmo se parte de una población inicial, que generalmente se crea de manera aleatoria. De esta población inicial se seleccionan los padres de la siguiente generación, para lo cual, usualmente se eligen los individuos de menor valor en la función de aptitud (para problemas de minimización) con una alta probabilidad. Sin embargo, también se da cabida a soluciones de mayor valor (soluciones malas), aunque con menor probabilidad, para permitir la exploración en diferentes zonas de la región factible y dar diversidad al algoritmo de búsqueda. El operador genético más importante es el cruce (MORILLO; MORENO y DÍAZ, 2014). La mutación solo cumple un papel secundario aportando, casualmente, cambios aleatorios para permitir la exploración de la búsqueda. Luego, se asignan parejas de padres de manera aleatoria y se procede a usar los operadores genéticos previamente definidos para hallar nuevas soluciones (hijos). Después de la transferencia de genes y la generación de hijos se usa algún criterio de selección para elaborar la nueva generación. El proceso se repite hasta cumplir con un criterio de parada. Los algoritmos genéticos son muy utilizados para resolver problemas de planificación de proyectos. En (LANCASTER y OZBAYRAK, 2007) se describe detalladamente la implementación de estos algoritmos en este tipo de problemas.

Los AG han sido ampliamente utilizados para resolver problemas de programación de proyectos (**cervantes2010nuevos**). En cuanto al RCPSP los AG son el método más utilizado para su solución (**cervantes2010nuevos**).

2.4.3. Optimización de la Colonia de Hormigas (Ant Colony Optimization)

La idea en la que se sustenta este algoritmo fue planteada en (DORIGO, 1992). Su algoritmo de optimización es un metaheurístico de poblaciones que intenta reproducir el mecanismo utilizado por las hormigas para localizar el alimento e informar a la colonia dónde se encuentra. La ruta seguida por las hormigas tiene

la característica de ser la de menor distancia desde su nido hasta el alimento. Este algoritmo se cataloga como un procedimiento de construcción seudo-aleatoria (MORILLO; MORENO y DÍAZ, 2014).

Estableciendo una analogía con el comportamiento natural de las hormigas, inicialmente, estas se desplazan de manera aleatoria, dejando por cada camino recorrido una cierta cantidad de feromonas. Las siguientes hormigas se desplazarán aleatoriamente pero con mayor probabilidad de ir por los caminos que tengan mayor concentración de esta sustancia. Éstas, a su vez, dejan su propia feromona, la cual, con el tiempo, se evapora paulatinamente; es decir, aquellas rutas que no se visiten recurrentemente tienden a desaparecer. De esta manera, algunos caminos se vuelven más atractivos por su alta concentración de feromona.

El principio de este algoritmo consiste en que la probabilidad de que una hormiga seleccione una ruta está condicionada por el número de hormigas que la hayan seleccionado previamente y de su distancia (FRANCISCO, 2002). Paulatinamente, las hormigas preferirán usar las rutas más cortas. El concepto de mínima distancia entre el nido y la fuente de alimento se adaptará a la función de aptitud, cuyo mínimo valor indicará la ruta a seleccionar. En este algoritmo se construyen soluciones iterativamente.

En (MORILLO; MORENO y DÍAZ, 2014) se define la probabilidad de incorporar el elemento j de la solución elaborada hasta el elemento i, de la siguiente manera:

$$P_{i,j} = \begin{cases} \frac{[\tau_{i,j}(t)]^{\alpha} \cdot [\eta_{i,j}]^{\beta}}{\sum_{u \notin M_k} [\tau_{iu}(t)]^{\alpha} \cdot [\eta_{iu}]^{\beta}} & \text{si } j \notin M_k \\ 0 & \text{en otro caso} \end{cases} \tag{2.4.1}$$

donde:

$\eta_{i,j}$: parámetro que representa una cierta "visibilidad", relacionada con la función de aptitud.

$\tau_{i,j}(t)$: rastro de feromona, que depende del tiempo.

α: importancia relativa de la feromona.

β: importancia relativa de la "visibilidad".

M_k: memoria asociada a la hormiga k.

En (MERKLE; MIDDENDORF y SCHMECK, 2002) se desarrolló por primera vez una aplicación de la optimización de la colonia de hormigas al RCPSP. Bajo este enfoque, una sola hormiga se define como una aplicación completa para elaborar una solución factible. En cada iteración, la siguiente actividad a seleccionar depende del valor promedio del tiempo de inicio más tardío de cada actividad. Dicho promedio será la feromona que representa el aprendizaje y el efecto de las anteriores hormigas (MORILLO; MORENO y DÍAZ, 2014).

2.4.4. Optimización basada en cúmulos de partículas (PSO)

Los algoritmos de optimización basados en cúmulos de partículas o Particle Swarm Optimization (PSO) están inspirados en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. El algoritmo PSO mantiene un conjunto de soluciones, también llamadas partículas, que son inicializadas aleatoriamente en el espacio de búsqueda. Cada partícula posee una posición y velocidad

que cambia conforme avanza la búsqueda. En el movimiento de una partícula influye su velocidad y las posiciones donde la propia partícula y las partículas de su vecindario encontraron buenas soluciones. En el contexto de PSO, el vecindario de una partícula se define como un conjunto de partículas del cúmulo. El vecindario de una partícula puede ser global, en el cual todas las partículas del cúmulo se consideran vecinas, o local, en el que sólo las partículas más cercanas se consideran vecinas (J. F. CHICANO, 2007).

2.5. Generación de Soluciones del Problema de Planificación con Recursos Limitados

EL Esquema Generador se Secuencias (Schedule Generator Scheme, SGS por sus siglas en Inglés) es una metodología que determina la manera de seleccionar cada actividad para elaborar una secuencia factible, asignando tiempo de inicio y de finalización a las actividades.

Los SGS son el núcleo de la mayoría de los procedimientos heurísticos para la solución del RCPSP (KOLISCH y S. HARTMANN, 1999). Los SGS construyen una solución factible extendiendo paso a paso una secuencia parcial de actividades, que inicialmente asigna el tiempo de inicio igual cero a la primera actividad planificada de la secuencia. Una secuencia parcial es aquella donde únicamente un subconjunto de las n actividades de un proyecto han sido planificadas (MEDRANO BROCHE, 2012).

Los esquemas de generación de secuencias pueden aplicarse en dos direcciones: la programación hacia adelante, es decir cuando se comienza programando la actividad ficticia inicio y por último la actividad ficticia fin. Sin embargo, las actividades de una instancia de RCPSP pueden ser programadas en orden inverso, es decir se comienza con la actividad ficticia fin, programando gradualmente todas las actividades intermedias y se termina cuando se asigna tiempo de inicio a la actividad ficticia de inicio. Esto se puede hacer fácilmente invirtiendo todas las relaciones de precedencia, es decir los sucesores de la actividad se convierten en sus predecesores y los predecesores en sucesores.

Existen dos esquemas diferentes, en serie SSGS y en paralelo Esquema Generador de Soluciones en Paralelo (por sus siglas en inglés, Parallel Schedule Generator Scheme) (PSGS). Ambos esquemas permiten encontrar soluciones factibles para el RCPSP.

- Esquema en Serie: Este esquema propuesto por Kelley (1963) y descrito en (KOLISCH, 1996), usa un número de etapas igual al número de actividades del proyecto. En cada etapa, se selecciona una actividad según una regla de prioridad y se programa tan pronto como sea posible, siempre y cuando cumpla las restricciones de precedencias y de recursos. El algoritmo termina cuando se hayan programado todas las actividades. En (ibíd.) se asegura que el conjunto de soluciones generadas por este algoritmo siempre contendrá la solución óptima.

- Esquema en Paralelo: Este esquema, descrito en (ibíd.), consiste en programar tantas actividades como sea posible en cada instante en el tiempo, respetando las restricciones de recursos y de precedencias. Este tiempo se denomina tiempo de decisión o nivel, y está asociado con el tiempo de finalización

de alguna actividad en ejecución. La definición de este esquema exige que no exista ningún período tal que una actividad que sea factible, en relación a las restricciones de recursos y precedencias, no sea programada. Aunque estos esquemas en paralelo pueden producir secuencias más compactas, no siempre contienen la solución óptima, pues las actividades programadas en cada nivel pertenecen al primer subconjunto del conjunto potencia, y el óptimo puede no pertenecer a ese subconjunto.

2.5.1. Esquema Generador de Soluciones en Serie

El SSGS consiste en $g = 0, 1, 2, ...$ estados, en cada uno de ellos las actividades de un proyecto son planificadas sin violar las restricciones de recursos y las relaciones de precedencia. Asociado a cada estado g existen dos conjuntos disjuntos de actividades. Un conjunto $D_g = \left\{ j \in J \setminus S_g : A_j(t) \subseteq S_g \right\}$ de las actividades que pueden ser seleccionadas para planificarse, donde $A_j(t)$ es el conjunto de las actividades antecesoras de la actividad $j \in J$ planificadas en el instante $t = 1, \ldots, T$. El otro conjunto S_{ec_g}, es el de las actividades que ya han sido planificadas en el estado g. Se tiene que $\bar{R}_k(t) = |R_k| - U_k(S, t)$ es el número de recursos del tipo $k \in K$, disponibles en el instante t, y que $F_g = \left\{ f_j : j \in S_g \right\}$ es el conjunto de los tiempos de fin de todas las actividades que se han planificado.

Algoritmo 1 Esquema Generador de Soluciones en Serie (SSGS)

1: INITIALIZE: $g = 0, Sec_0 = \{1\}$
2: **while** $S_g \neq J$ **do**
3: $g = g + 1$
4: CALCULATE: $D_g, F_g, \bar{R}_k(t)$ TAL QUE $t \in F_g, k \in K$
5: SELECT: $j \in D_g$
6: CALCULATE: $es_j = \max\{f_h\}, h \in A_j$
7: CALCULATE: $s_j = \max\left\{ t : t \geq es_j, r_{jk} \leq \bar{R}_k(\tau), \forall k \in K, \tau \in \left[t, t + d_j\right] \cap F_g \right\}, h \in A_j$
8: CALCULATE: $f_j = s_j + d_j$
9: INSERT: $Sec_g = Sec_{g_{-1}} \cup \{j\}$
10: $C_{max} = max_j\left(f_j\right), j \in J$
11: **end while**

2.6. Trabajos relacionados

2.6.1. Planificación de proyectos

En (CHANG; CHRISTENSEN y ZHANG, 2001) se aborda el problema de asignar empleados a tareas teniendo en cuenta sus habilidades, su salario y su dedicación con el objetivo de minimizar el coste y la duración del proyecto.

Alba y Chicano (ALBA y J. Francisco CHICANO, 2005), (J. FRANCISCO CHICANO, 2007) abordan este mismo problema analizando distintos proyectos automáticamente generados con un generador de

instancias.

La asignación de work packages (WP) de un proyecto a equipos de programadores con el objetivo de reducir la duración del proyecto es un problema de optimización que ha sido abordado en (GIULIO ANTONIOL y HARMAN., 2004). Los mismos autores amplían el problema en (ANTONIOL; PENTA y HARMAN, 2004) para optimizar también la asignación de programadores a equipos y considerar soluciones robustas que sean capaces de tolerar el posible abandono de WPs, errores, revisiones e incertidumbre en las estimaciones.

Una rama específica de la Investigación Operativa se ha concentrado en resolver los problemas específicos de optimización de redes de proyectos, es decir, de encontrar secuencias o programaciones (schedule) que, compatibles con los requisitos tecnológicos y económicos, optimizan uno (o varios) valores representativos de una magnitud relevante del proyecto, sea esta el coste, la duración total o algún otro. En general, todos los problemas de este tipo son NP-Duro por lo que los métodos exactos de solución sólo suelen ser útiles en problemas de escasa entidad (VICTORIO, 2008).

Una parte de la investigación se ha centrado en la producción de algoritmos para la obtención de soluciones aproximadas para estos problemas. La cantidad de modelos y propuestas producidas es muy importante. Esta producción se ha visto enriquecida en las últimas décadas con las aportaciones de la Inteligencia Artificial, que ha proporcionado nuevos métodos heurísticos para la obtención de soluciones de calidad, así como técnicas, como la propagación de restricciones (constraint propagation), para acotar el espacio de búsqueda (ibíd.).

2.6.2. Algoritmos Genéticos

Existen varios trabajos donde se relaciona la metaheurística GA con el problema RCPSP (LANCASTER y OZBAYRAK, 2007). Todos usan los operadores básicos de los GA: selección, mutación y cruzamiento. Las variantes de utilización de estos operadores son muy diferentes, así como la definición de la función objetivo, pero todas coinciden en que con el uso de este algoritmo se obtienen buenos resultados.

Un ejemplo de la relación entre las restricciones que plantea el RCPSP y la codificación del GA se observa en (NARVÁEZ MOLINA, 2010). En la codificación cada valor del gen en el cromosoma representa el número de la tarea y la posición del gen dentro de la cadena especifica el orden de ejecución de dichas tareas. En este se obtiene una planificación de las tareas donde solo se tiene en cuenta la restricción impuesta por la precedencia entre actividades. Por lo que, una vez obtenida la mejor solución parcial, se describe otro procedimiento no heurístico para determinar los tiempos de inicio y de fin de las tareas del cronograma. Este enfoque, aunque simplifica el tiempo de cómputo, compromete su efectividad en proyectos con más de 30 tareas.

Por otra parte en (ZOULFAGHARI; NEMATIAN; MAHMOUDI y KHODABANDEH, 2013) se presenta un GA que se utiliza para resolver el RCPSP a gran escala y mejorar las soluciones. En este trabajo la codificación se realiza a través de cromosomas en cuyos genes se representan los tiempos de inicio de las actividades, se asume que la posición del gen dentro de la cadena hace referencia al número de la actividad.

Los operadores de cruzamiento y mutación tienen en cuenta tanto las restricciones de precedencia como la disponibilidad de recursos. Los resultados mostrados con este procedimiento arrojan buenas soluciones para proyectos de hasta 30 tareas o más de 120 tareas. Sin embargo, para proyectos de aproximadamente 60 actividades se coloca en desventaja con respecto a otros enfoques de solución.

Algo interesante se plantea en los trabajos publicados por (FRANCO, 2013; HARTMANN, 1997). En ambos se emplea un SGS para obtener los cromosomas de la población inicial. Esta característica acelera la convergencia del algoritmo (GIL LONDOÑO, 2006). Los operadores de cruzamiento utilizados son de un punto y de dos puntos. El algoritmo propuesto en (FRANCO, 2013) presenta mejores resultados con el operador de cruzamiento de dos puntos, sin embargo el tiempo de cómputo es considerablemente mayor que su variante de un solo punto.

2.7. Conclusiones del capítulo

- Como resultado de la investigación se obtiene una completa revisión del estado del arte respecto a las metodologías iterativas de aproximación para la solución del RCPSP.
- Se evidenció que se han utilizado diversos métodos metaheurísticos para la resolución del RCPSP, y que actualmente los más estudiados para este fin son los heurísticos basados en poblaciones.
- Se elige la metaheurística GA, ya que se pudo constatar que es uno de los algoritmos más utilizados y reconocidos para este fin. Este algoritmo evita, a través del diseño de sus operadores, la convergencia prematura hacia una solución óptima local.
- Se pudo apreciar que los SGS son el núcleo de la mayoría de los procedimientos heurísticos para la solución del RCPSP, pues su uso garantiza la construcción de buenas soluciones en un tiempo relativamente corto.
- Se valora el empleo de un SSGS para generar la población inicial del GA, basado en la experiencia descrita en los trabajos revisados.

CAPÍTULO 3

ALGORITMO GENÉTICO PARA LA PLANIFICACIÓN

3.1. Introducción

En el capítulo se presenta el algoritmo genético para la planificación automática del cronograma de un proyecto de software con recursos limitados. El algoritmo se ha estructurado siguiendo las acciones más utilizadas identificadas en los procedimientos analizados en el capítulo anterior. La propuesta está enfocada en obtener automáticamente las tres mejores planificaciones de un cronograma de proyecto de software, optimizando el tiempo total estimado para la duración del proyecto y realizando de forma eficiente la asignación de recursos humanos, en cuanto al aprovechamiento de los mismos en la resolución de las actividades, buscando un adecuado balance entre ellos.

3.2. Descripción de la organización

El algoritmo que se propone ha sido diseñado pensando en que la organización en estudio, se dedica al desarrollo de productos de software. Dicha organización está formada por proyectos. Cada proyecto está compuesto por actividades y recursos humanos. Los recursos humanos a los que en su conjunto se le denomina equipo de desarrollo son las personas requeridas para ejecutar las actividades de un proyecto. Cada miembro del equipo de desarrollo está especializado en actividades específicas en el desarrollo del producto. En función de ello cada uno tiene un rol asociado, una evaluación del rol que está asociada al desempeño del mismo en cuanto a las competencias que encierra y un fondo de tiempo en horas que representa la cantidad de tiempo que puede dedicar al desarrollo de las tareas del proyecto.

3.3. Estructura general de la propuesta

Los objetivos principales en la gestión de entornos de proyectos, son la productividad de los recursos y la velocidad de los proyectos. El problema reside en que las decisiones que benefician uno de dichos objetivos suelen perjudicar al otro. Para que los proyectos sean realizados lo más rápido posible se debería disponer

de más recursos, pero entonces algunos de ellos estarían en parte ociosos. En cambio, si se quisiera saturar a los recursos serían los proyectos los que deberían esperar.

En función de lo anterior, la propuesta está encaminada a brindar una herramienta útil que pueda ser utilizada en la Fase de Planificación del proyecto. Su uso posibilitará la obtención de las tres mejores planificaciones de un cronograma de proyecto de software, donde se ha optimizado el tiempo total de duración del proyecto y la asignación de recursos humanos ha sido eficiente en cuanto al aprovechamiento de los mismos en la resolución de las actividades, buscando un adecuado balance entre ellos.

La herramienta obtenida es para el uso de recursos humanos con el rol de planificador, el cual debe trabajar de conjunto con el líder de proyecto para buscar su aprobación en cuanto a la selección de la mejor solución de planificación, acorde con las características del proyecto en cuestión. Para poder utilizar la herramienta el planificador debe tener habilidades de comunicación y trabajo en equipo, ser organizadotener y poseer conocimientos mínimos de gestión de proyectos. Es importante destacar también que debe tener conocimiento de los elementos pactados en la Fase de Inicio del proyecto donde se establecen los límites de tiempo y recursos.

La estructura general de la propuesta es como se muestra en 3.1:

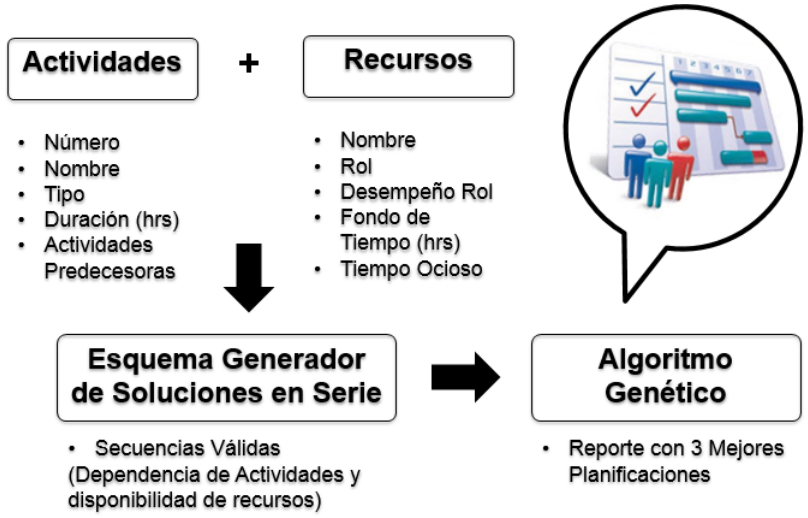


Figura 3.1. Estructura general de la propuesta (elaboración propia)

3.3.1. Datos de Entrada

El SSGS necesita como entrada a su base de datos los siguientes elementos:

- Conjunto de las actividades del proyecto, de cada una de ellas se debe especificar número, nombre, tipo, la duración en horas, el conjunto de actividades predecesoras y los recursos por rol que necesita.
- Conjunto de recursos humanos a utilizar en el proyecto, de cada uno de ellos se debe conocer nombre, rol y el fondo de tiempo en horas del que dispone durante la ejecución del proyecto.

- Fecha de Inicio y Fin del proyecto.

Ya con el almacenamiento de estos datos se procede a la ejecución del SSGS, donde se debe entrar la cantidad de secuencias de planificación a obtener. Luego estas secuencias de planificación obtenidas se almacenan en un fichero con formato .txt el cual será cargado por el GA, ya que las soluciones generadas serán su población inicial, se especifica además el nombre del proyecto en cuestión y el número de generaciones con las que trabajará dicho algoritmo.

3.3.2. Datos de Salida

El algoritmo genético diseñado genera, como resultado fundamental, secuencias de planificación optimizadas. Este resultado se recoge en un reporte con formato PDF que va a contener las tres mejores soluciones que obtenga el GA diseñado. El reporte está conformado por:

- En la cabecera se coloca el nombre del algoritmo desarrollado.
- Se hace alusión al proyecto que da origen al reporte.
- Se muestran las mejores soluciones obtenidas. El formato de una solución está compuesto por una secuencia de tareas, los recursos asignados a cada tarea y el tiempo de finalización total del proyecto.

3.3.3. Requerimientos de uso

El sistema se ha desarrollado para plataformas de escritorio. Para su correcto funcionamiento se debe contar como mínimo con una PC Intel Pentium 4 o superior, CPU 1.6 GHz o superior, 1 Gb de RAM, teniendo en cuenta el nivel de procesamiento necesario que depende de la cantidad de tareas que contengan los proyectos.

3.4. Conceptualización de la propuesta

La propuesta elaborada para la planificación debe contar con tres elementos fundamentales. El primer elemento es el proyecto que es descompuesto por actividades. Un segundo elemento, es el equipo de desarrollo que ejecuta el proyecto y un tercero son las decisiones tácticas y operativas a llevar a cabo en la planificación. Teniendo en cuenta cada uno de estos elementos se describen a continuación.

3.4.1. Proyecto

El proyecto es representado por un grafo dirigido sin ciclo $G = (J, E)$ donde J es el conjunto de los nodos (actividades) y $E = \left\{(i, j) : i \in A_j\right\}$ el conjunto de los arcos (relaciones de precedencia), donde $A_j \subset J$ es el conjunto de las actividades antecesoras de j y por ende tienen que ejecutarse antes que esta.

Las actividades son aquellas tareas que se deben completar para terminar el proyecto. El conjunto de actividades se denota como $J = \{j : j = 0, 1, ..., n\}$ donde $j = 0$ y $j = n$, son actividades ficticias (nodos inicio y fin), tienen tiempo de duración igual a cero, no requieren recursos para su procesamiento, solo indican cuando comienza y cuando termina el proyecto. Cada una de las j actividades del proyecto tal que $j \in J \setminus \{0, n\}$ pueden ser de $k \in K$ tipos, donde $K = \{1, 2, 3..m\}$ ($k = 1$ análisis, $k = 2$, diseño, $k = 3$ implementación, $k = m$ prueba). Las actividades $j \in J$ de tipo $k \in K$ requieren recursos con el rol correspondiente al tipo de actividad para que esta pueda ejecutarse. Cada actividad $j \in J$ tiene una duración d_j en horas (peso del arco). Las relaciones de precedencia serán del tipo *Fin-Inicio,* con $FS_{ij} = 0$, como espacio de tiempo máximo entre el fin de una actividad $i \in A_j$ y el inicio de una actividad j, donde A_j es el conjunto de actividades antecesoras de la actividad j.

3.4.2. Equipo de Desarrollo

Otro componente esencial son los miembros del equipo de desarrollo. Se tendrá en cuenta la categoría de recursos renovables, donde $R = \bigcup_{k \in K} R_k$ es el conjunto de todos los recursos disponibles del proyecto. Cada recurso r_k tiene un rol que está asociado a la especialización de ejecutar actividades de un tipo $k \in K$ (por ejemplo $R_1 = \{1, 2, ..., N_{R_1}\}$ es el grupo de los analistas del proyecto). Cada recurso r_k tiene un valor λ que es la evaluación del desempeño del rol $\lambda \in \{1, 2, ...6\}$ por tanto se dice que el recurso $r_{k\lambda}$ es especialista en realizar actividades del tipo k y tiene una capacidad productiva λ ($\lambda = 6$ alta, $\lambda = 4$ media, $\lambda = 2$ baja). Cada recurso tiene además un fondo de tiempo f_k y un tiempo ocioso o_k ambos en horas. El primero va a representar la cantidad de tiempo que puede dedicar al desarrollo de las tareas del proyecto, y por tanto para poder ejecutar una tarea j que tiene una duración d_j debe cumplirse que t_k sea mayor o igual a d_j. Por su parte el segundo representa el tiempo que ha estado desocupado desde el último estado g donde completó una tarea j hasta el estado g en que se le asigne otra tarea.

3.4.3. Desiciones tácticas

Las decisiones tácticas en el entorno del proyecto son tomadas en la Fase de Inicio teniendo en cuenta la planificación de capacidad de la organización. Esto se traduce en la elección de cuándo se debe comenzar el proyecto, de forma tal que exista una cantidad adecuada de recursos para ejecutar sus actividades y un presupuesto para enfrentar el mismo. Relacionada con la elección de la fecha de inicio del proyecto también está la definición de la fecha de compromiso (DDi). Esta decisión está condicionada por la capacidad productiva de la organización y por las características del proyecto.

Una vez decidido el inicio y fin del proyecto, así como el presupuesto a utilizar, estos datos serán utilizados por el algoritmo, donde se descartarán aquellas planificaciones que excedan el tiempo pactado ya que no cumplen el objetivo de la investigación que es minimizar el tiempo de desarrollo. En cuanto al presupuesto, como no se varía la duración de las actividades, ni la cantidad de recursos asignados a las mismas, cualquiera de las propuestas de planificación obtenidas que estén dentro de la fecha de compromiso, no afectará el presupuesto pactado.

3.4.4. Desiciones operativas

Las decisiones a nivel operativo son las relacionadas con la secuenciación de las actividades y con la asignación de los recursos que las ejecutan. Concretamente se deben tomar dos decisiones. La primera consiste en determinar la estrategia empleada para priorizar las actividades del proyecto que pueden ser planificadas en un instante de tiempo t. La segunda decisión está conformada por la estrategia de asignar los recursos disponibles para que ejecuten las actividades.

Asignación de prioridad a las actividades

La estrategia que se tuvo en cuenta para calcular el valor de prioridad π_j de cada actividad j, fue por la regla de prioridad número de sucesores inmediatos (MTS), el valor de la prioridad es calculado por $\pi_j = |Suc_j|$ donde Suc_j es el conjunto de las actividades sucesoras inmediatas de la actividad j. Se planifica primero la actividad con mayor valor de π_j.

Asignación de prioridad a los recursos

El valor de prioridad de cada recurso disponible de tipo k en cada instante t depende de que el recurso posea el rol asociado al tipo de la actividad j, un fondo de tiempo (en horas) que debe ser mayor a la duración de la actividad a planificar, así como el tiempo ocioso, que representa el tiempo que el recurso se encuentra sin realizar ninguna labor.

3.5. Generación de soluciones factibles

Los SGS son el núcleo de la mayoría de los procedimientos heurísticos para la solución del RCPSP (KOLISCH y S. HARTMANN, 1999). A continuación se describe cómo se realiza la representación de las soluciones y la adaptación del SGS para la obtención de soluciones factibles del problema planteado.

3.5.1. Representación de las soluciones

Los SGS construyen una solución factible extendiendo paso a paso una secuencia parcial de actividades, que inicialmente asigna el tiempo de inicio igual cero a la primera actividad planificada de la secuencia. Una secuencia parcial es aquella donde únicamente un subconjunto de las n actividades de un proyecto ha sido planificada (MEDRANO BROCHE, 2012). Existen dos esquemas diferentes, en serie SSGS y en paralelo PSGS.Ambos esquemas permiten encontrar soluciones factibles para el RCPSP.

Según (KOLISCH, 1996), SSGS construye soluciones activas, tiene mejor ejecución ante instancias grandes y el conjunto de soluciones generadas por este algoritmo siempre contendrá la solución óptima si consideramos únicamente el problema no restringido en cuanto a recursos. En cambio PSGS genera soluciones sin retraso, el conjunto de soluciones sin retraso es un subconjunto de las soluciones activas por lo que tiene, en general, un cardinal menor. Pero tiene una desventaja importante, y es que puede no contener

ninguna solución óptima para una medida regular, algo que no ocurre con el conjunto de las soluciones activas. Por ello en este trabajo se abordará el SSGS.

3.5.2. SSGS para el RCPSP

El SSGS consiste en $g = 0, 1, 2, ...$ estados, en cada uno de ellos las actividades de un proyecto son planificadas sin violar las restricciones de recursos y las relaciones de precedencia. Asociado a cada estado g existen dos conjuntos disjuntos de actividades. Un conjunto $D_g = \left\{ j \in J \setminus S_g : A_j(t) \subseteq S_g \right\}$ de las actividades que pueden ser seleccionadas para planificarse, donde $A_j(t)$ es el conjunto de las actividades antecesoras de la actividad $j \in J$ planificadas en el instante $t = 1, \ldots, T$. El otro conjunto S_{ec_g}, es el de las actividades que ya han sido planificadas en el estado g. Se tiene que $\bar{R}_k(t) = |R_k| - U_k(S, t)$ es el número de recursos del tipo $k \in K$, disponibles en el instante t, y que $F_g = \left\{ f_j : j \in S_g \right\}$ es el conjunto de los tiempos de fin de todas las actividades que se han planificado.

Algoritmo 2 Esquema Generador de Soluciones en Serie (SSGS)

1: INITIALIZE: $g = 0, Sec_0 = \{1\}$
2: **while** $S_g \neq J$ **do**
3: $g = g + 1$
4: CALCULATE: $D_g, F_g, \bar{R}_k(t)$ TAL QUE $t \in F_g, k \in K$
5: SELECT: $j \in D_g$
6: CALCULATE: $es_j = \max\{f_h\}, h \in A_j$
7: CALCULATE: $s_j = \max\left\{ t : t \geq es_j, r_{jk} \leq \bar{R}_k(\tau), \forall k \in K, \tau \in \left[t, t + d_j\right] \cap F_g \right\}, h \in A_j$
8: CALCULATE: $f_j = s_j + d_j$
9: INSERT: $Sec_g = Sec_{g_{-1}} \cup \{j\}$
10: $C_{max} = max_j\left(f_j\right), j \in J$
11: **end while**

3.6. Algoritmo genético para el RCPSP

Los Algoritmos Genéticos son uno de los métodos aproximados que mejores resultados proporcionan al resolver los problemas de programación de proyectos con recursos limitados en problemas que no pueden ser resueltos por los métodos exactos (POSADA, 2010). En (LANCASTER y OZBAYRAK, 2007) se describe detalladamente la implementación de estos algoritmos en este tipo de problemas.

Los algoritmos genéticos son el enfoque que obtiene los mejores resultados en la resolución de este problema. Se utiliza una codificación de los individuos basada en una lista de actividades y se utilizan los esquemas serie y paralelo para generar la programación del proyecto (POSADA, 2010). En esta investigación se pretende desarrollar un algoritmo genético para la planificación de tareas y asignación automática de recursos en un proyecto de desarrollo de software.

3.7. Descripción del algoritmo genético

Existen varias representaciones de las soluciones del RCPSP, en especial a partir de la incorporación de las técnicas metaheurísticas como alternativa de resolución. Estas representaciones condicionan las técnicas que se van a poder emplear dentro de un algoritmo basado en una de ellas (FRANCISCO, 2002). En la presente investigación se diseña un GA que maneja la planificación de proyecto como una lista de actividades. Una lista de actividades es una permutación de actividades $\lambda = (j_1, j_2, \ldots, j_n)$ válida respecto a las relaciones de precedencia, o sea, a cada actividad le corresponde una posición u orden mayor que el de cualquiera de sus predecesoras. De este modo, si $i = j_P$ diremos que la actividad i está en la posición p, o que el orden de i en λ es p. Se tiene además que $j_1 = 1$ y $j_n = n \; \forall \lambda$. Habitualmente la población inicial de un GA se obtiene a partir de un conjunto de individuos generados al azar, sin embargo en esta investigación se propone como población inicial un grupo de soluciones válidas generadas a partir del SSGS descrito en 1. En los pocos trabajos que abordan el uso de alguna técnica heurística o de optimización local en la obtención de la población inicial, se constata que esta inicialización no aleatoria puede acelerar la convergencia del algoritmo (GIL LONDOÑO, 2006).

3.7.1. Operadores del algoritmo

Codificación

En los GA, y en general en todos los procedimientos metaheurísticos, la forma en la que se representan las soluciones es crucial para el comportamiento de los mismos. Los diversos autores que han desarrollado algoritmos para resolver este problema han propuesto diferentes codificaciones para las soluciones, que suelen ser generalmente una extensión o adaptación de la codificación utlizada para la versión estándar o único modo del problema.

La codificación de la planificación de un proyecto se representa a través de una secuencia de planificación I (individuo), con una lista de valores enteros (cromosoma), donde cada uno de sus elementos (genes) hace referencia a una actividad. La dimensión de I coincide con la cantidad de actividades del proyecto en cuestión. La lista de actividades definida en 3.7.1 representa una planificación de las actividades del caso de estudio de la Figura 1.1a. Obsérvese que en este caso, por ejemplo, la actividad 3 se encuentra en la posición 4, o lo que es lo mismo, el orden de 3 en I es 4.

$$I = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \left[\begin{matrix} 1 & 2 & 4 & 3 & 5 & 6 & 7 \end{matrix} \right] \end{array} \tag{3.7.1}$$

Selección basada en Ruleta

Una parte fundamental del funcionamiento de un AG es el proceso de selección de los candidatos a generar los nuevos individuos de la población. Se debe garantizar que los mejores individuos tengan una

mayor posibilidad de ser padres (reproducirse) frente a los individuos menos buenos. Mientras más apto sea un organismo (valor de aptitud más alto), más veces será seleccionado para reproducirse. Sin embargo, los individuos menos adaptados también deben tener probabilidades de reproducirse, debido a que su material genético puede ser útil para encontrar buenas soluciones (POSADA, 2010). El operador de selección establece cuántas copias se crearán de cada individuo en la siguiente generación (VELASCO, 2002).

El método de la ruleta ha sido el método más comúnmente utilizado desde el origen de los GA. Los padres se seleccionan de acuerdo con su aptitud. Los individuos mejores son los que tienen mayores posibilidades de ser elegidos, siendo la probabilidad de ser seleccionados proporcional a su aptitud (POSADA, 2010).

En este caso se emplea el operador de selección estocástica con remplazo, para lo cual se crea una ruleta 3.2 con tantas casillas como secuencias de planificación (individuos) tenga la población; el ángulo de cada casilla es proporcional a la probabilidad de supervivencia. Se juega a la ruleta tantas veces como individuos tenga la población y se crea una copia de cada individuo ganador. De este modo se mantiene la diversidad entre los individuos de la población y se evita la convergencia prematura del algoritmo.

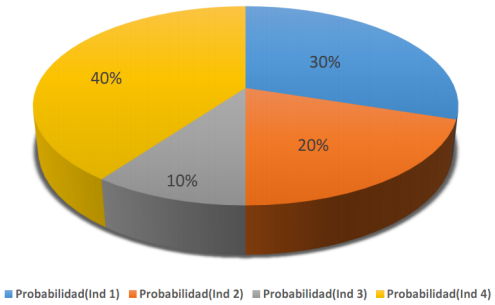


Figura 3.2. Ruleta con operador de probabilidad proporcional

Este mecanismo de selección utiliza una ordenación creciente atendiendo a la aptitud o calidad de las secuencias de planificación para después seleccionar atendiendo a su posición en dicha ordenación. Esto reduce el problema de la disminución en la diversidad de la población debido a valores de aptitud similares.

Cruzamiento en un punto

En los sistemas biológicos, el cruce es un proceso complejo entre parejas de cromosomas. Estos cromosomas se alinean, luego se fraccionan en ciertas partes y posteriormente intercambian fragmentos entre sí. Este operador permite el intercambio de información entre las secuencias de la población actual, por lo tanto tiene un gran impacto en el resultado del algoritmo. Es por ello que es uno de los operadores genéticos que más atención ha recibido por parte de los investigadores (ibíd.).

La técnica del cruce en un punto fue propuesta por Davis. En un cromosoma de longitud n existen $n-1$ puntos de cruce, por lo que el punto de cruce k será un número aleatorio en el intervalo $[0, n-1]$. El hijo tendrá los genes del 0 al k iguales a los de su padre y los genes desde $k+1$ hasta n heredados en el orden

relativo de su madre. Este procedimiento asegura que la secuencia generada sea factible. La hija se genera de manera análoga (POSADA, 2010).

Para el proceso de cruce se elige la generación de dos nuevas secuencias de planificación (hijos) a partir de dos secuencias de planificación de la actual generación (padres). El operador que se emplea fue propuesto en (NARVÁEZ MOLINA, 2010), es probabilístico y resulta ser una variación del operador de cruzamiento por un punto. Con su utilización se garantiza que las nuevas secuencias obtenidas también cumplan las restricciones de precedencia de actividades impuestas por el proyecto en cuestión. El procedimiento para su utilización es el que se describe a continuación:

- Se genera un número aleatorio u entre 1 y n, el cual representa el punto de cruce entre las secuencias de planificación.
- El primer hijo hereda los genes del padre de izquierda a derecha hasta u, completando la secuencia genética con los genes de la madre, de izquierda a derecha, que aún no formen parte de la misma.
- De igual modo, el segundo hijo hereda los genes de la madre de izquierda a derecha hasta u, y completa la secuencia genética heredando los genes del padre, de izquierda a derecha, que aún no formen parte de la misma.

En 3.7.2 se muestra cómo se efectúa el cruce entre dos secuencias de planificación que codifican soluciones válidas para el caso de estudio propuesto en la Figura 1.1a. Nótese que el valor aleatorio u para el punto de cruce es 2.

$$
\begin{array}{rc}
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
Padre = & \left[\begin{array}{ccccccc} 1 & 2 & 4 & 3 & 5 & 6 & 7 \end{array}\right] \\
 & \\
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
Madre = & \left[\begin{array}{ccccccc} 1 & 4 & 5 & 2 & 3 & 6 & 7 \end{array}\right] \\
 & \\
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
Hijo1 = & \left[\begin{array}{ccccccc} 1 & 2 & 4 & 5 & 3 & 6 & 7 \end{array}\right] \\
 & \\
 & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
Hijo2 = & \left[\begin{array}{ccccccc} 1 & 4 & 2 & 3 & 5 & 6 & 7 \end{array}\right]
\end{array}
\tag{3.7.2}
$$

Mutación

La mutación juega un papel fundamental en cualquier GA debido a que brinda la posibilidad de salir de un entorno de búsqueda monótono y pasar a otro que probablemente tenga entre sus habitantes mejores soluciones.

Algunos investigadores sugieren usar porcentajes altos de mutación al comienzo de la búsqueda y luego decrementarlos exponencialmente. Sin embargo lo más frecuente es que el valor se fije al inicio del algoritmo y se usen probabilidades de mutación inferiores al 5 por ciento (POSADA, 2010).

El operador propuesto en este trabajo se basa en el descrito por (NARVÁEZ MOLINA, 2010), al igual que el diseñado para el cruce, es probabilístico y garantiza que la nueva secuencia de planificación que se obtenga cumpla con las restricciones de precedencia impuestas por el proyecto. Los pasos para lograr la mutación son los siguientes:

- Generar un número aleatorio u entre 1 y $n-1$.
- Si la actividad ubicada en el orden u de la secuencia de planificación no es predecesora de la actividad situada en la posición $u+1$ de la misma secuencia, entonces se intercambian ambas actividades. Caso contrario volver al paso anterior.

Este simple procedimiento se ejemplifica en 3.7.3 para el primer hijo obtenido en 3.7.2. En este caso el valor aleatorio u es 4.

$$Hijo1 = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \begin{bmatrix} 1 & 2 & 4 & 5 & 3 & 6 & 7 \end{bmatrix}$$

$$Hijo1Mutado = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 & 7 \end{bmatrix} \tag{3.7.3}$$

Función Objetivo

Un aspecto importante en los algoritmos genéticos es la determinación de la calidad o fortaleza de los individuos en la población, precisamente la función objetivo es la que determina la aptitud del individuo.

Debido a la complejidad del problema, el hecho de generar una secuencia que sea factible ya consiste en si mismo en un problema complejo. Por ello, los algoritmos deben admitir soluciones no factibles y por tanto las funciones objetivo incluyen factores de penalización (POSADA, 2010).

Cuando se manejan únicamente soluciones factibles en cuanto a relaciones de precedencia y uso de recursos no renovables, la minimización de la duración del proyecto es la función objetivo del algoritmo

(POSADA, 2010). Dada la naturaleza del RCPSP, la función objetivo empleada en esta investigación es minimizar el tiempo total de duración del proyecto. Su representación se plantea a continuación:

$$min f_n \tag{3.7.4}$$

Sujeto a:

$$f_n = \max\left\{f_j, \forall f_j \in F_g\right\} \qquad f_n < DD_i \tag{3.7.5}$$

Para obtener el tiempo total de planificación del proyecto de cualquier secuencia de planificación se ha utilizado una variante del SSGS, descrita en el Algoritmo 3. En este caso se omite la generación del conjunto de las actividades elegibles D_g para cada estado. Esto es debido a que ya se cuenta con la secuencia de actividades planificadas, denotada por P, por lo que solo es necesario encontrar los tiempos de finalización de cada una de las actividades. Además, se asignan los recursos necesarios para realizar las actividades, a partir de una estrategia de selección que prioriza aquellos recursos que llevan más tiempo sin realizar alguna actividad.

Algoritmo 3 Calcular Función Objetivo

1: INITIALIZE: $g = 0, Sec_0 = \{1\}$
2: **while** $S_g \neq J$ **do**
3: $g = g + 1$
4: CALCULATE: $F_g, \bar{R}_k(t)$ TAL QUE $t \in F_g, k \in K$
5: SELECT: $j \in P$
6: CALCULATE: $es_j = \max\{f_h\}, h \in A_j$
7: CALCULATE: $s_j = \max\left\{t : t \geq es_j, r_{jk} \leq \bar{R}_k(\tau), \forall k \in K, \tau \in \left[t, t + d_j\right] \cap F_g\right\}, h \in A_j$
8: CALCULATE: $f_j = s_j + d_j$
9: CALCULATE: R_{kj}
10: UPDATE LAST JOB: R_{kj}
11: INSERT: $Sec_g = Sec_{g_{-1}} \cup \{j\}$
12: $C_{max} = max_j\left(f_j\right), j \in J$
13: **end while**

Reemplazo

El reemplazo de individuos es el operador que permite introducir a los hijos generados por los operadores genéticos en la población. En los GA existen dos maneras de hacerlo: reemplazando toda la población o reemplazando solo unos cuantos individuos (ibíd.).

El Elitismo es uno de los operadores no convencionales de los GA. Es de los métodos más utilizados

para mejorar la convergencia de los GA. Su principal objetivo es mantener la mejor solución al problema de una generación a otra. Este operador primeramente almacena un grupo de individuos de la población inicial. Este conjunto de individuos es reemplazado a medida que aparecen otros que tengan mejores aptitudes. Esto evita que se pierdan los individuos que tienen mejor aptitud en el GA.Este operador consiste básicamente en incorporar una élite de r miembros de la población i a la población $i + 1$, sin pasar por la población intermedia.

El elitismo puede mejorar el funcionamiento de los GA al evitar que se pierda la mejor solución entre dos iteraciones consecutivas (ARRANZ DE LA PEÑA y PARRA TRUYOL, 2014). Comúnmente, el tamaño de la élite es bastante pequeño (1 ó 2 para $n = 50$) para evitar la convergencia prematura. En esta investigación el tamaño para la élite se ha fijado en 1.

3.8. Pseudocódigos

Algoritmo 4 Cruzamiento, de la clase OperadorReproduccionCrucePunto

```
1:  size := Population->Generation->Size()
2:  for i:=0,2,4,... ,size do
3:      father := Population->Generation->data(i)
4:      mother := father->partner
5:      random := Random(1,n)
6:      for j:=0,1,2,...,u do
7:          child1 := father->Genome->data(j)
8:          child2 := mother->Genome->data(j)
9:      end for
10:     for k:=0,1,2,...,n do
11:         if Contains(child1, mother->Genome->data(k)) == false then
12:             child1->Add(mother->Genome->data(k))
13:         end if
14:         if Contains(child2, father->Genome->data(k)) == false then
15:             child2->Add(father->Genome->data(k)
16:         end if
17:     end for
18:     father := child1
19:     mother := child2
20: end for
```

3.9. Conclusiones del capítulo

- La codificación seleccionada para el GA propuesto es apropiada para representar el problema abordado.

Algoritmo 5 Optimización, de la clase Algoritmo Genético

```
1:  BeginOptimization()
2:  repeat
3:      procedure IterateOptimization
4:          Selection()
5:          AssignPartner()
6:          Crosover()
7:          Mutation()
8:          Elitism()
9:          AssignProbability()
10:         FindElite()
11:         generation := generation + 1
12:         UpdateMeasures()
13:     end procedure
14: until Stop() == true
```

- El uso de las secuencias de planificación obtenidas con el SSGS, como población inicial del GA propuesto permitió concentrar la búsqueda en un área de soluciones prometedoras.
- La función objetivo seleccionada permitió evaluar la calidad del individuo a partir de la secuencia de planificación representada en su codificación.
- El uso de operadores probabilísticos de cruzamiento y mutación garantizaron de forma eficiente explorar el espacio de soluciones válidas.
- El reporte con formato PDF diseñado muestra las mejores secuencias de planificación obtenidas como resultado del algoritmo implementado, lo que va a contribuir a la toma de desiciones en la ejecución de los proyectos de software.

CAPÍTULO 4

APLICACIÓN DEL ALGORITMO Y ANÁLISIS DE RESULTADOS

4.1. Introducción

Se analizan los resultados obtenidos de la aplicación del algoritmo genético propuesto. Para ello se emplean instancias de la biblioteca PSPLib.

Conjuntos de Problemas de PSPLib

La biblioteca PSPLib (http://www.om-db.wi.tum.de/psplib) contiene conjuntos de problemas para los distintos PSP, así como soluciones óptimas y heurísticas. Los conjuntos de datos pueden ser utilizados para la evaluación de los procedimientos de solución de los RCPSP. Los investigadores pueden descargar la referencia fija para evaluar sus algoritmos. Además, pueden enviar sus resultados para ser añadidos a la biblioteca (KOLISCH y SPRECHER, 1997). A continuación se describe el formato general de los archivos que propone PSPLib como instancias de problemas:

- Número de actividades (jobs).
- Número de modos en que cada actividad puede ejecutarse (#modes).
- Número de tipos de recursos renovables existentes en el problema (renewable).
- Disponibilidad máxima de cada tipo de recurso renovable (RESOURCEAVAILABILITIES).
- Número de tipos de recursos no renovables existentes en el problema (nonrenewable).
- Número de sucesores de cada actividad (#successors).
- Conjunto de actividades sucesoras de cada actividad (successors).
- Duración de cada actividad (duration).

- Número de recursos necesarios para cada actividad (R1, R2, R3, R4)

Cada prueba tiene diez réplicas por lo que para J30, J60 y J90 hay 480 instancias y 600 instancias para el conjunto J120. Existen cuatro recursos renovables diferentes.

4.2. Aplicación de la propuesta para una instancia de 30 actividades

La instancia de prueba *j301-1* obtenida de PSPLib, contiene 30 tareas y 41 recursos, estos últimos están clasificados en cuatro tipos. En la Figura 4.1 se observa el formato del fichero *j301-1.sm*.

Tabla 4.1. Resultados para la instancia de prueba *j301-1*

n=30		**Tamaño de la Población**				
	Sol	**20**	**40**	**60**	**80**	**100**
Generaciones	**25**	55	50	53	52	46
	50	55	47	51	49	45
	75	54	50	47	53	47
	100	53	53	54	49	45

En la Tabla 4.1 y la Figura A.1 se observa el mejor resultado obtenido por el GA implementado, atendiendo al número de individuos de la población y la cantidad de generaciones.

```
jobs (incl. supersource/sink ):  32
RESOURCES
  - renewable                 :  4   R
  - nonrenewable              :  0   N
****************************************
PRECEDENCE RELATIONS:
jobnr.    #modes  #successors   successors
   1        1          3           2  3  4
   2        1          3           6  11 15
   3        1          3           7  8  13
   4        1          3           5  9  10
   5        1          1           20
   6        1          1           30
   7        1          1           27
   8        1          3           12 19 27
   9        1          1           14
  10        1          2           16 25
  11        1          2           20 26
  12        1          1           14
  13        1          2           17 18
  14        1          1           17
  15        1          1           25
  16        1          2           21 22
  17        1          1           22
  18        1          2           20 22
  19        1          2           24 29
  20        1          2           23  25
  21        1          1           28
  22        1          1           23
  23        1          1           24
  24        1          1           30
  25        1          1           30
  26        1          1           31
  27        1          1           28
  28        1          1           31
  29        1          1           32
  30        1          1           32
  31        1          1           32
  32        1          0
```

```
****************************************
REQUESTS/DURATIONS:
jobnr. mode duration  R 1  R 2  R 3  R 4
----------------------------------------
  1      1     0       0    0    0    0
  2      1     8       4    0    0    0
  3      1     4      10    0    0    0
  4      1     6       0    0    0    3
  5      1     3       3    0    0    0
  6      1     8       0    0    0    8
  7      1     5       4    0    0    0
  8      1     9       0    1    0    0
  9      1     2       6    0    0    0
 10      1     7       0    0    0    1
 11      1     9       0    5    0    0
 12      1     2       0    7    0    0
 13      1     6       4    0    0    0
 14      1     3       0    8    0    0
 15      1     9       3    0    0    0
 16      1    10       0    0    0    5
 17      1     6       0    0    0    8
 18      1     5       0    0    0    7
 19      1     3       0    1    0    0
 20      1     7       0   10    0    0
 21      1     2       0    0    0    6
 22      1     7       2    0    0    0
 23      1     2       3    0    0    0
 24      1     3       0    9    0    0
 25      1     3       4    0    0    0
 26      1     7       0    0    4    0
 27      1     8       0    0    0    7
 28      1     3       0    8    0    0
 29      1     7       0    7    0    0
 30      1     2       0    7    0    0
 31      1     2       0    0    2    0
 32      1     0       0    0    0    0
****************************************
RESOURCEAVAILABILITIES:
  R 1  R 2  R 3  R 4
   12   13    4   12
****************************************
```

Figura 4.1. Fichero j301-1.sm obtenido de PSPLib

Se observa que para esta instancia de 30 tareas el algoritmo converge a 45 unidades de tiempo. Un valor muy cercano al 43 propuesto en PSPLib como mejor solución para este caso. Por otra parte, la información anterior indica que los valores más cercanos a la solución óptima se logran con poblaciones de 40 o más individuos, después de las 25 generaciones.

Otras pruebas realizadas con esta instancia arrojaron como resultado que para una misma población inicial de n individuos, si se asigna un número de generaciones entre 25 y 100, el GA obtiene valores iguales o muy similares.

4.3. Aplicación de la propuesta para una instancia de 60 actividades

La instancia de prueba *j601-1* (Ver Figura 4.4), obtenida de PSPLib, cuenta con 60 tareas y contiene 49 recursos distribuidos en cuatro tipos. En la Tabla 4.2 y la Figura A.2 se observa el mejor resultado obtenido por el GA implementado, atendiendo al número de individuos de la población y la cantidad de generaciones.

Tabla 4.2. Resultados para la instancia de prueba *j601-1*

n=60		Tamaño de la Población				
	Sol	**20**	**40**	**60**	**80**	**100**
Generaciones	**25**	83	83	94	82	78
	50	90	83	82	83	77
	75	89	86	83	80	78
	100	89	87	90	78	77

Para esta instancia de 60 tareas, el algoritmo obtiene como mejor valor 77 unidades de tiempo, que coincide con la mejor solución propuesta por PSPLib. Las mejores soluciones se obtienen con poblaciones iniciales de 80 o más individuos, a partir de las 25 generaciones.

Al ejecutar el GA para esta instancia, con una misma población inicial de n individuos, si se varía el número de generaciones en el rango de 25 a 100, se obtiene el mismo valor o valores muy similares. Esto no se puede afirmar en caso de cambiar la población inicial para una misma cantidad de individuos y un mismo número de generaciones.

4.4. Aplicación de la propuesta para una instancia de 120 actividades

La instancia de prueba *j1201-1*, obtenida de PSPLib, cuenta con 120 tareas y contiene 48 recursos distribuidos en cuatro tipos. En la Tabla 4.3 se observa el mejor resultado obtenido por el GA implementado, atendiendo al número de individuos de la población y la cantidad de generaciones.

Tabla 4.3. Resultados para la instancia de prueba *j1201-1*

n=120		Tamaño de la Población						
	Sol	**20**	**40**	**60**	**80**	**100**	**120**	**140**
Generaciones	**25**	146	141	144	143	136	134	126
	50	143	138	143	141	139	136	118
	75	149	140	145	138	131	126	110
	100	140	144	141	137	134	115	114

Para esta instancia de 120 tareas, el algoritmo obtiene como mejor valor 110 unidades de tiempo. Un valor muy cercano al 105 propuesto en PSPLib como mejor solución para este caso. Las mejores soluciones se obtienen con poblaciones iniciales de 140 o más individuos, a partir de las 25 generaciones.

Al ejecutar el GA para esta instancia, con una misma población inicial de n individuos, si se varía el número de generaciones en el rango de 25 a 100, se obtiene el mismo valor o valores muy similares. Esto no se puede afirmar en caso de cambiar la población inicial para una misma cantidad de individuos y un mismo número de generaciones.

Resultados de la comparación entre casos de estudio

Luego de analizar por separado los tres casos, se evidencia que para una misma población inicial, independientemente del número de generaciones entre 25 y 100, el GA implementado obtiene soluciones iguales o muy similares.

Por otra parte, si se varía la población inicial se observa que para la instancia de 30 tareas la convergencia se obtiene con poblaciones de 40 o más individuos. Por su parte para la instancia de 60 tareas las mejores soluciones se alcanzan con mayor regularidad a partir de poblaciones de 80 individuos o más, mientras que para la instancia de 120 tareas las mejores soluciones se alcanzan con mayor regularidad a partir de poblaciones de 140 individuos. En los tres casos se consideran estos resultados con un número de generaciones igual o superior a 25.

Lo anterior evidencia que, a medida que aumenta el número de tareas de un proyecto, el GA precisa de un número mayor de individuos en la población para obtener las mejores soluciones.

En los diferentes problemas modelo de la PSPLib se observa que algunos problemas presentan mayor complejidad que otros. Esto puede observarse de diferentes formas en los resultados: por un lado entre mayor sea la diferencia entre la solución óptima y el valor de alguna cota inferior, podría pensarse que mayor es la complejidad del problema. Dicha complejidad puede estar relacionada con la cantidad de precedencias entre las actividades y en mayor proporción con la disponibilidad y el consumo de los recursos. Esto lleva a pensar que realizando en el futuro un análisis profundo de la estructura de cada problema podría dar un indicio de qué tan fácil o difícil podría llegar a ser un problema determinado incluso antes de intentar resolverlo, para así controlar el valor de los parámetros de búsqueda en forma automática, dependiendo de la complejidad del problema.

Convergencia del Algoritmo Genético

En la Figura 4.2 se realiza el análisis de los resultados obtenidos con la ejecución del GA para la instancia *j301-1*. En este se considera el valor de la función objetivo del mejor individuo de la población en el intervalo de generaciones de 0 a 100, con paso 5. Se representan los resultados de cinco corridas del algoritmo, cada una de ellas varía el tamaño de la población: 20, 40, 60, 80 y 100 individuos respectivamente.

En todos los casos se constata que el ritmo de convergencia del algoritmo es mayor en las primeras 40 generaciones que en las restantes. En estas últimas se mejora la solución únicamente en los casos de 60 y 100 individuos. En los cinco casos el GA implementado converge al valor óptimo o uno bastante cercano al óptimo, en un número pequeño de generaciones.

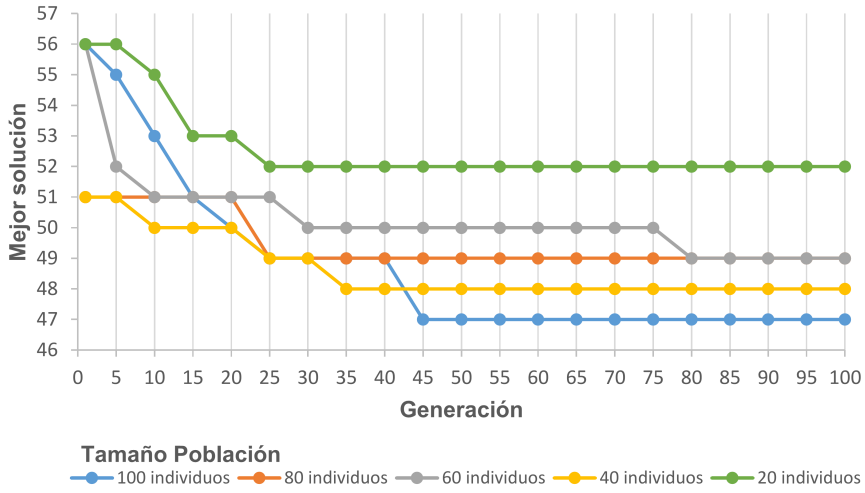


Figura 4.2. Convergencia del GA

Balance de carga entre recursos

La información representada en la Figura 4.3 corresponde a la instancia *j301-1* y se obtiene a partir de la ejecución del GA implementado. Se muestra el por ciento que representa el tiempo ocioso de cada recurso respecto al total de los tiempos de ejecución de cada una de las actividades que solicitan el empleo de recursos de su mismo tipo. En esta instancia de 30 tareas se cuenta con 4 tipos de recursos, en la gráfica se observa un favorable balance de carga entre los recursos de un mismo tipo.

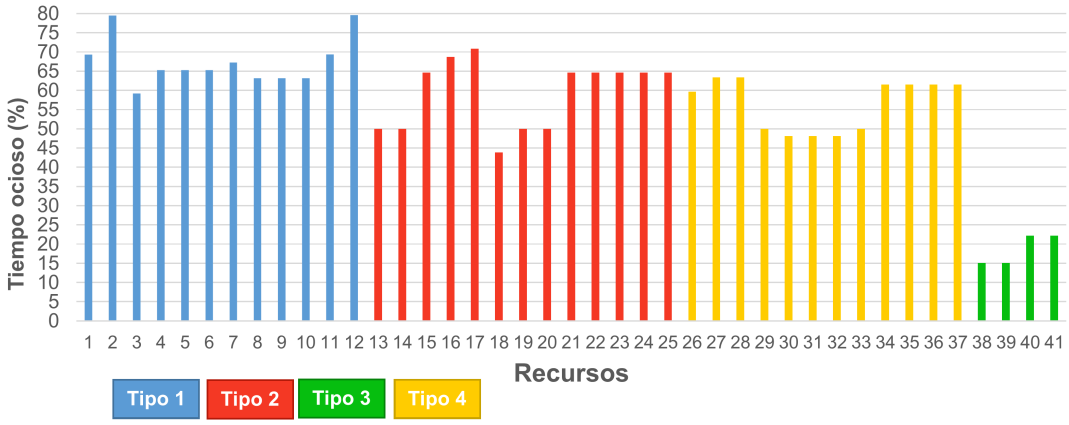


Figura 4.3. Balance de carga entre recursos del proyecto

```
jobs (incl. supersource/sink ):  62
RESOURCES
  - renewable                 :  4   R
  - nonrenewable              :  0   N
******************************************
PRECEDENCE RELATIONS:
jobnr.    #modes  #successors   successors
   1        1          3           2   3   4
   2        1          3           5  10  15
   3        1          3           7  14  29
   4        1          3           8  12  16
   5        1          3           6  22  24
   6        1          2          17  38
   7        1          1          23
   8        1          3           9  20  40
   9        1          2          13  35
  10        1          2          11  45
  11        1          3          26  37  44
  12        1          2          21  27
  13        1          1          18
  14        1          3          18  19  34
  15        1          2          25  59
  16        1          2          55  58
  17        1          2          32  43
  18        1          2          28  33
  19        1          1          28
  20        1          2          27  31
  21        1          1          39
  22        1          1          31
  23        1          1          51
  24        1          1          48
  25        1          1          40
  26        1          2          49  54
  27        1          2          30  51
  28        1          2          47  61
  29        1          2          41  57
  30        1          1          56
  31        1          1          43
  32        1          1          57
  33        1          1          39
  34        1          1          44
  35        1          2          36  39
  36        1          1          42
  37        1          1          58
  38        1          2          50  60
  39        1          1          53
  40        1          1          53
  41        1          1          46
  42        1          1          48
  43        1          2          51  58
  44        1          1          50
  45        1          1          53
  46        1          1          56
  47        1          1          52
  48        1          1          55
  49        1          1          60
  50        1          1          61
  51        1          1          60
  52        1          1          54
  53        1          1          56
  54        1          1          55
  55        1          1          57
  56        1          1          61
  57        1          1          59
  58        1          1          59
  59        1          1          62
  60        1          1          62
  61        1          1          62
  62        1          0
```

```
REQUESTS/DURATIONS:
jobnr. mode duration  R 1  R 2  R 3  R 4
------------------------------------------
  1      1     0       0    0    0    0
  2      1     8      10    0    0    0
  3      1     1       0    1    0    0
  4      1    10       0    9    0    0
  5      1     6       0    4    0    0
  6      1     5       0    0    0    1
  7      1     8      10    0    0    0
  8      1     9       0    0    6    0
  9      1     1       0    0    0    8
 10      1     9       0    6    0    0
 11      1     8       0    0    0    3
 12      1     3       0    7    0    0
 13      1     6       8    0    0    0
 14      1     2       0    0    0    1
 15      1     5       0    0    0    9
 16      1     1       6    0    0    0
 17      1     3       2    0    0    0
 18      1    10       0    0    2    0
 19      1     9       7    0    0    0
 20      1     1       5    0    0    0
 21      1     3       0    0    8    0
 22      1     6       0    4    0    0
 23      1     3       0    0    4    0
 24      1     3       0    5    0    0
 25      1     7       0    0    1    0
 26      1     6       9    0    0    0
 27      1    10       0    7    0    0
 28      1     9       3    0    0    0
 29      1     8       0    0    3    0
 30      1     4       0    0    7    0
 31      1     3       6    0    0    0
 32      1     3       0    0    0    4
 33      1     6       0    7    0    0
 34      1     1       0    0    0    4
 35      1     9       0    0    1    0
 36      1     9       9    0    0    0
 37      1     1       0    7    0    0
 38      1     2       5    0    0    0
 39      1     4       0    0    1    0
 40      1     9       0    0    0    5
 41      1    10       0    0    0    1
 42      1     8       0    0    0    9
 43      1     4       0    0    6    0
 44      1     3       0    0    0    1
 45      1     6       0    0    0    9
 46      1     6       0    0    0    7
 47      1     7       4    0    0    0
 48      1     3       0    8    0    0
 49      1     2       0    2    0    0
 50      1    10       0    0    7    0
 51      1     4       0    5    0    0
 52      1     2       2    0    0    0
 53      1     1       0    1    0    0
 54      1     4       0    0    6    0
 55      1    10       0    0    0    7
 56      1     8       0    0    3    0
 57      1     6       0    4    0    0
 58      1    10       0    0    9    0
 59      1     3       0    0    0    7
 60      1    10       0    3    0    0
 61      1     1       0    0    0    1
 62      1     0       0    0    0    0
******************************************
RESOURCEAVAILABILITIES:
  R 1  R 2  R 3  R 4
   13   11   12   13
```

Figura 4.4. Fichero j601-1.sm obtenido de PSPLib

4.5. Análisis del impacto económico de la propuesta

El análisis económico de la presente investigación está basado en el costo del Algoritmo genético para optimizar la planificación del cronograma de un proyecto de software teniendo en cuenta: el desarrollo del algoritmo genético, el costo estimado para su integración con la herramienta GESPRO 13.05 y el costo estimado para la implantación de la propuesta en la Red de Centros de la UCI. Para el análisis de los costos se deben tener en cuenta los costos tangibles e intangibles, elementos tales como: el costo de electricidad de las PC, costo de conectividad a internet, costo de los locales que se utilizan para el desarrollo e integración de la propuesta, costo de los locales donde se impartieron las clases de capacitación, el costo asociado al salario de los profesores que impartieron las clases de capacitación, Insumos Informáticos, materiales de oficina, entre otros. En la presente investigación se analizó solo el costo asociado al salario del personal que trabajó en el desarrollo, así como el personal estimado para la integración e implantación de la propuesta. Para la obtención de los costos asociados al salario del personal se tuvo en cuenta los siguientes conceptos:

- Fondo salarial de un trabajador: Se tiene a partir de la plaza que ocupa, cargo que desempeña, categoría docente, categoría científica, años de experiencias docentes y Pago Adicional, así como el descuento del 5 % de seguridad social.
- Tarifa horaria: Se obtiene a partir del fondo salarial del trabajador y la Resolución 8 del 2005.
- Costo Total: Total de horas del tiempo dedicado del trabajador * tarifa horaria.

4.5.1. Análisis del Costo Asociado al desarrollo de la propuesta

El costo de desarrollo del algoritmo genético para la planificación está asociado al salario de las personas implicadas en la organización, planificación, ejecución, control y seguimiento del desarrollo del mismo. Estuvieron inmersa en esta actividad 2 especialistas. El desarrollo de la propuesta estuvo constituido por 2 periodos, siendo cada uno iterativo e incremental. A continuación se muestra el costo por períodos desde septiembre 2013 hasta julio del 2015. El costo total fue de $ 23 212.8.

4.5.2. Análisis del costo estimado para la implantación de la propuesta en un proyecto de un centro de desarrollo

La implantación del algoritmo genético para optimizar la planificación del cronograma de un proyecto de software requiere de un conjunto de acciones que permitan obtener un resultado exitoso. Para estimar el costo de implantación de la propuesta en la UCI, se asumirá que se realizará la implantación en al menos un proyecto de cada uno de los centros de desarrollo que tiene la universidad. De acuerdo a estos datos se muestra en la Tabla 4.4 el comportamiento del costo que tendría la implantación de la propuesta en un proyecto de un centro de desarrollo, siguiendo el conjunto de acciones que se especifican. Se debe destacar que para su implantación no se requiere de una gran cantidad de recursos económicos (computadoras, hojas

Período	Trabajadores			Tiempo Dedicado			Costo Total
	Profesor	Fondo Salarial	Tarifa Horaria	Horas al Mes	Cantidad de Meses	Total de Horas	
Septiembre de 2013 - Julio de 2014	1	731	3,8	172	9	1548	$ 5882,4
	2	1005	5,3	120	9	1080	$5724
	Total						**11 606, 4**
Septiembre de 2014 - Julio de 2015	1	731	3,8	172	9	1548	$ 5882,4
	2	1005	5,3	10	9	1080	$5724
	Total						**11 606, 4**

Figura 4.5. Costo de desarrollo de la propuesta por periodos

y bolígrafos), siendo el tiempo del recurso determinante para calcular el costo de implantación. En la estimación se utilizó una tarifa horaria de $ 5.97 por hora de trabajo de cada persona involucrada. Esta tarifa horaria responde al salario más alto de las personas involucradas, en este caso el de un Jefe de Departamento con categoría docente de Instructor y con una evaluación de Adecuado en su desempeño trimestral. En la etapa de planificación se propone que participe la persona que se encuentra capacitada para realizar la implantación de la propuesta y el Jefe del Proyecto donde se va a implantar, donde se deben revisar todos los aseguramientos para garantizar el éxito del proceso. Los directivos a participar en el encuentro inicial, en el informe de los resultados y que deben participar también en el resto de las actividades son: Jefe de Departamento, Jefe de Proyecto y Planificador del proyecto. En la capacitación teórica, la capacitación práctica y el análisis de resultados deben participar además de los directivos, el personal involucrado en las tareas de planificación y análisis que se consideren necesarios. Con este análisis de puede concluir que el costo de implantación de la propuesta en la UCI es relativamente bajo respecto al impacto que puede tener la aplicación de la propuesta en los proyectos de desarrollo. Se destaca también que se estimó con el salario más alto posible para el personal involucrado e incorporando cuatro personas a capacitar en cada proyecto, números que pueden ser menores si la dirección del centro de desarrollo donde se implante lo considera pertinente. Con la incorporación de un módulo a la herramienta GESPRO que permita el cálculo de los puntos de función siguiendo una o las dos generalizaciones detalladas en esta investigación, se minimiza el esfuerzo y el tiempo que se le dedica a la aplicación de la propuesta.

Tabla 4.4. Estimación del costo de aplicación en un proyecto de un centro de desarrollo

Acción	Tiempo (h)	Personal	Costo ($)
Planificación para la implantación	4	2	47,76
Encuentro inicial con directivos	2	3	35,82
Capacitación teórica	3	4	71,64
Capacitación práctica	2	4	47,76
Análisis de los resultados	2	4	47,76
Informe de los resultados	2	5	59,70
Costo total para un proyecto			**$ 310,44**

4.5.3. Estimación del costo de desarrollo de la propuesta para la herramienta GESPRO

Para incluir este algoritmo genético a la herramienta, contando con la experiencia de los especialistas del Departamento de Investigaciones de Gestión de Proyecto que mantienen esta herramienta, se estima que el costo de desarrollo sea como se muestra en la Tabla 4.5. Este costo de desarrollo se considera relativamente bajo también. Al poder realizar, en todos los centros de desarrollo de la universidad, la planificación del proyecto utilizando el algoritmo genético propuesto, a través de la herramienta los beneficios que se reportarán serán mayores. En la estimación se utilizó una tarifa horaria de $ 4.15 por hora de trabajo de cada persona involucrada, teniendo en cuenta el salario que percibe un especialista de un centro de desarrollo de la UCI. De igual manera que en la implantación de la propuesta, el desarollo de este algoritmo no requiere de una gran cantidad de recursos económicos (computadoras, hojas y bolígrafos), siendo el tiempo el recurso determinante para calcular el costo de desarrollo.

Tabla 4.5. Estimación del costo de desarrollo para la herramienta GESPRO

Fases	Tiempo (h)	Personal	Costo ($)
Análisis y Diseño	16	1	66,40
Desarrollo	24	2	199,20
Pruebas	8	1	33,20
Costo Total de desarrollo			**$ 298,80**

Con este análisis se puede concluir que el costo de desarrollo de la propuesta para su inclusión en la herramienta GESPRO es relativamente bajo respecto al impacto que puede tener su aplicación en los proyectos de desarrollo de la UCI. A partir de las estimaciones realizadas se obtiene como costo para la utilización en la UCI, un total de 609,24 pesos.

La utilización de esta propuesta permitirá realizar una planificación más exacta del proceso de desarrollo de software en el proyecto, lo que contribuirá a utilizar con mayor eficiencia los recursos del proyecto al planificar con mayor exactitud sus necesidades dentro del proceso de desarrollo. Además de proporcionar una información más precisa para calcular los costos asociados al desarrollo del software, lo que ayudará a la alta gerencia del proyecto a tomar las decisiones que se necesiten en cada momento.

4.6. Análisis del impacto social, lineamiento de la política económica y social del partido y la Revolución de la propuesta

La propuesta realizada en la presente investigación se encuentran en correspondencia con los Lineamientos de la Política Económica y Social del Partido y la Revolución aprobados en el VI Congreso del Partido Comunista de Cuba para actualizar el modelo económico cubano garantizando la continuidad del Socialismo (PCC, 2011). Los lineamientos que se apoyan con las propuestas realizadas en la investigación son el 7, 8, 16, 41, 45, 73, 84, 131 y 132. Los mismos se muestran a continuación.

- **Lineamiento 7**: Lograr que el sistema empresarial del país esté constituido por empresas eficientes, bien organizadas y eficaces, y serán creadas las nuevas organizaciones superiores de dirección empresarial. Se desarrollará la cooperación entre las empresas para garantizar mayor eficiencia y calidad.
- **Lineamiento 8**: El incremento de facultades a las direcciones de las entidades estará asociado a la elevación de su responsabilidad sobre la eficiencia, eficacia y el control en el empleo del personal, los recursos materiales y financieros que manejan; unido a la necesidad de exigir la responsabilidad a aquellos directivos que con decisiones, acciones u omisiones ocasionen daños y perjuicios a la economía.
- **Lineamiento 16**: Las empresas deciden y administran su capital de trabajo e inversiones hasta el límite previsto en el plan.
- **Lineamiento 41**: Una relación entre el crecimiento de la productividad del trabajo y del ingreso medio de los trabajadores, que no deteriore el equilibrio monetario interno ni la eficiencia de la economía nacional.
- **Lineamiento 73**: Trabajar con el máximo rigor para aumentar la credibilidad del país en sus relaciones económicas internacionales, mediante el estricto cumplimiento de los compromisos contraídos.
- **Lineamiento 84**: Garantizar la sostenibilidad del ciclo de producción de los renglones exportables y diseñar la organización de los esquemas correspondientes para esto.
- **Lineamiento 132**: Perfeccionar las condiciones organizativas, jurídicas e institucionales para establecer tipos de organización económica que garanticen la combinación de investigación científica e

innovación tecnológica, desarrollo rápido y eficaz de nuevos productos y servicios, su producción eficiente con estándares de calidad apropiados y la gestión comercializadora interna y exportadora, que se revierta en un aporte a la sociedad y en estimular la reproducción del ciclo. Extender estos conceptos a la actividad científica de las universidades.

Los lineamientos de la Política Económica y Social del Partido y la Revolución, sometidos a debate y respaldado por la mayoría de nuestro pueblo definen que el sistema económico que prevalecerá continuará basándose en la propiedad socialista de todo el pueblo sobre los medios fundamentales de producción. La política económica del Partido se corresponderá con el principio de que sólo el socialismo es capaz de vencer las dificultades y preservar las conquistas de la Revolución, y que en la actualización del modelo económico primará **la planificación**, la cual tendrá en cuenta las tendencias del mercado. Todos y cada uno de los lineamientos relacionados con la temática fueron tenidos en cuenta y sentaron pauta de alguna forma en el desarrollo del trabajo.

4.7. Conclusiones del capítulo

- Las instancias de prueba de PSPLib que se analizan en este capítulo demuestran que con el algoritmo implementado se obtienen secuencias de planificación óptimas o cercanas a las óptimas para la solución del problema abordado.
- Los secuencias de planificación obtenidas van a permitir disminuir los tiempos de desarrollo en los proyectos y contribuyen al apoyo de la toma de decisiones.
- El costo de la propuesta fue de 23 511,6 pesos cubanos (CUP) a partir del costo asociado al desarrollo, e integración de la misma en la herramienta GESPRO.

Conclusiones

Con realización de la presente investigación se arribaron a las siguientes conclusiones:

- El estudio de las metaheurísticas existentes para la optimización de problemas existentes ofrecen oportunidades para mejorar el proceso de planificación en la Gestión de Proyectos de Software.
- Con el GA diseñado se obtienen secuencias de planificación cuasi óptimas para el RCPSP.
- La creación de secuencias de planificación con el SSGS, como población inicial para el GA, permite la convergencia del mismo en un número pequeño de generaciones y se logra además un adecuado balance de carga entre los recursos de un mismo tipo.
- Con el análisis de las instancias de prueba de PSPLib de 30, 60 y 120 actividades respectivamente se demostró que a medida que aumenta el número de tareas de un proyecto, el GA precisa de un número mayor de individuos en la población para obtener las mejores soluciones.

Recomendaciones

Con interés de buscar mayor eficiencia en el proceso de planificación de tareas de un proyecto de software se proponen las siguientes recomendaciones:

- Analizar el uso de otros operadores que pudieran ser implementados en el GA para comprobar si mejoran las soluciones obtenidas, así como introducir el concepto de múltiples modos de procesamiento.
- Comprobar el desempeño del algoritmo ante proyectos con un volumen de más de 120 actividades para verificar la escalabilidad del mismo.
- Desarrollar un mecanismo de asignación de recursos que, además de tener en cuenta el tiempo ocioso, utilice otros indicadores. Ejemplo: desempeño del rol.
- Integrar el algoritmo propuesto a la herramienta GESPRO para validar de forma práctica la propuesta en proyectos de software en la UCI.

Acrónimos

AI Inteligencia Artificial. 1

AON Actividad En Nodo (por sus siglas en inglés, Activity On Node). 17

GA Algoritmos Genéticos (por sus siglas en inglés, Genetic Algorithms). 3, 6, 26, 27, 31, 32, 35, 39, 40, 42–45, 47–51, 58, 59

GESPRO Suite de Gestión de Proyectos. 2, 55, 57

JIT Justo a Tiempo (por sus siglas en inglés, Just In Time. 19

OR Investigación de Operaciones. 1

PSGS Esquema Generador de Soluciones en Paralelo (por sus siglas en inglés, Parallel Schedule Generator Scheme). 29, 37

PSP Problema de Planificación de Proyectos (por sus siglas en inglés, Project Scheduling Problem). 5, 9, 16–19, 22, 46

PSPLib Biblioteca de Problemas de Planificación de Proyectos. 5, 6, 25, 46–50, 57, 58

RCPSP Problema de Planificación de Proyectos con Recursos Limitados (por sus siglas en inglés, Resource Constrained Project Scheduling Problem). 3–5, 9, 19–26, 28, 29, 31, 32, 37, 39, 43, 46, 58

SGS Esquema Generador de Soluciones (por sus siglas en inglés, Schedule Generator Scheme). 25, 29, 32, 37

SSGS Esquema Generador de Soluciones en Serie (por sus siglas en inglés, Series Schedule Generator Scheme). 6, 29, 30, 32, 34, 35, 37–39, 43, 45, 58

UCI Universidad de las Ciencias Informáticas. 2, 53–55, 59

Referencias bibliográficas

ABDEL-HAMID, Tarek K. y MADNICK, Stuart E. 1989. Lessons learned from modeling the dynamics of software development. *Commun. ACM*. 1989, vol. 32, págs. 1426-1438. Url: ⟨http://dx.doi.org/http://doi.acm.org/10.1145/76380.76383⟩. ISSN 0001-0782.

ABDEL-HAMID, T.K. 1989. The dynamics of software project staffing: a system dynamics based simulation approach. *Software Engineering, IEEE Transactions on*. 1989, vol. 15, n.º 2, págs. 109-119. Url: ⟨http://dx.doi.org/10.1109/32.21738⟩. ISSN 0098-5589.

ALBA, Enrique y CHICANO, J. Francisco. 2005. Management of software projects with GAs. En. *Management of software projects with GAs*. 2005.

ÁLVAREZ VÁZQUEZ, Álvaro. 2012. *Apuntes de la asignatura de Tema 3:Gestión de Proyectos*. 2012. Dirección: ⟨http://www.alumnos.unican.es/~uc17923/Tema3.pdf⟩.

ANTONIOL, Giuliano; PENTA, Massimiliano Di y HARMAN, Mark. 2004. A Robust Search-Based Approach to Project Management in the Presence of Abandonment, rework, error and uncertainty. 2004, págs. 172-183. Url: ⟨http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/METRIC.2004.1357901⟩. ISSN 1530-1435.

ARRANZ DE LA PEÑA, Jorge y PARRA TRUYOL, Antonio. 2014. Algoritmos Genéticos. *Universidad Carlos III*. 2014.

ARTIGUES, Christian; MICHELON, Philippe y REUSSER, Stéphane. 2003. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*. 2003, vol. 149, n.º 2, págs. 249-267. ISSN 0377-2217.

BAAR, Tonius; BRUCKER, Peter y KNUST, Sigrid. 1999. *Tabu search algorithms and lower bounds for the resource-constrained project scheduling problem*. 1999. ISBN 978-1-4615-5775-3.

BLAZEWICZ, J.; ECKER, K.; H., Pesch E.; G., Schmidt y WEGLARZ, Jan. 2007. *Handbook on Scheduling : From Theory to Applications*. 2007. ISBN 978-3-540-28046-0.

BROOKS, SP y MORGAN, BJT. 1995. Optimization using simulated annealing. *The Statistician*. 1995, págs. 241-257.

BRUCKER, Peter; DREXL, Andreas; MÖHRING, Rolf; NEUMANN, Klaus y PESCH, Erwin. 1999. Resource-constrained project scheduling: Notation, classification, models, and methods. *European journal of operational research*. 1999, vol. 112, n.º 1, págs. 3-41.

CAPETILLO CORBEA, Jarsey. 2012. *Implementación de un algoritmo híbrido basado en optimización por Enjambre de Partículas para el problema de planificación de proyecto con múltiples modos de procesamiento*. 2012.

CHANG, Carl K.; CHRISTENSEN, Mark J. y ZHANG, Tao. 2001. Genetic Algorithms for Project Management. *Annals of Software Engineering*. 2001, vol. 11, n.º 1, págs. 107-139. Url: ⟨http://dx.doi.org/10.1023/A:1012543203763⟩. ISSN 1022-7091.

CHANG, Carl K.; JIANG, Hsin-yi; DI, Yu; ZHU, Dan y GE, Yujia. 2008. Time-line based model for software project scheduling with genetic algorithms. *Information and Software Technology*. 2008, vol. 50, n.º 11, págs. 1142-1154. Url: ⟨http://www.sciencedirect.com/science/article/B6V0B-4S3G3VS-1/2/9991742cbb7d0d4aa05b880d83ca1ba5⟩. ISSN 0950-5849.

CHICANO, J. F. 2007. *Metaheurísticas e Ingeniería de Software*. 2007.

CUBAINDUSTRIA. 2014. *Normas Cubanas Online*. 2014. Dirección: ⟨http://www.nconline.cubaindustria.cu/.⟩.

DEBELS, Dieter; DE REYCK, Bert; LEUS, Roel y VANHOUCKE, Mario. 2006. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*. 2006, vol. 169, n.º 2, págs. 638-653.

DEMEULEMEESTER, Erik Leuven. 2002. *Project scheduling: a research handbook*. 2002. ISBN 1-40207-051-9.

DONGWON KANG, Jinhwan Jung y BAE, Doo-Hwan. 2011. Constraint-based human resource allocation in software projects. *Software: Practice and Experience*. 2011, vol. 41, n.º 5, págs. 551-577. Url: ⟨http://dx.doi.org/10.1002/spe.1030⟩. ISSN 1097-024X.

DORIGO, Marco. 1992. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*. 1992.

EDEKI, C. 2013. Agile Unified Process. International Journal of Computer Science and Mobile Applications. *International Journal of Computer Science and Mobile Applications*. 2013, vol. 1, págs. 13-17. ISSN 2321-8363.

FEO, Thomas A y RESENDE, Mauricio GC. 1989. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*. 1989, vol. 8, n.º 2, págs. 67-71.

FRANCISCO, B. 2002. *Nuevos métodos de resolucion del problema de secuenciacion de proyectos con recursos limitados*. 2002. ISBN 84-370-5566-0.

FRANCO, Gutiérrez. 2013. A Genetic Algorithm for the Resource Constrained Project Scheduling Problem (RCPSP). *School of Industrial Engineering, Universidad de La Sabana*. 2013, vol. 10, n.º 20. ISSN 1814-6333.

GARCÍA, Isaías Alvarez et al., 2002. *Planificación y desarrollo de proyectos sociales y educativos*. 2002. ISBN 968-18-5416-0.

GIL LONDOÑO, Natyhelem. 2006. Algoritmo genético. *Medellin, Colombia*. 2006.

GIULIO ANTONIOL, Massimiliano Di Penta y HARMAN., Mark. 2004. Search-Based Techniques for Optimizing Software Project Resource Allocation. En. *Genetic and Evolutionary Computation Conference GECCO 2004*. 2004, págs. 1425-1426. Lecture Notes in Computer Science. 10.1007/978-3-540-24855-2_162. Url: ⟨http://dx.doi.org/10.1007/978-3-540-24855-2_162⟩.

GLOVER, Fred. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*. 1986, vol. 13, n.º 5, págs. 533-549.

GLOVER, Fred. 1989. Tabu search. *ORSA Journal on computing*. 1989, vol. 1, n.º 3, págs. 190-206. ISSN 1526-5528.

GLOVER, Fred. 1998. A template for scatter search and path relinking. En. *Artificial evolution*. 1998, págs. 1-51.

HARTMANN. 1997. A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. *Institut fur Betriebswirtschaftslehre der Universitat Kiel, Germany*. 1997, vol. 10, n.º 20. ISSN 1520-6750.

HARTMANN, Sonke y BRISKORN, Dirk. 2010. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*. 2010, vol. 207, n.º 1, págs. 1-14. Url: ⟨http://www.sciencedirect.com/science/article/B6VCT-4XNN5G6-2/2/2500b7b287428c1b2f4823dd73df7d17⟩. ISSN 0377-2217.

HARTMANN y BRISKORN, Dirk. 2008. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*. 2008, vol. 207, n.º 02/2008, págs. 1-14.

HOLLAND, J. 1975. *Adaptation in Natural and Artificial Systems*. 1975. ISBN 9780262082136.

HOLLAND, John H. 1992. Genetic Algorithms. *Scientific American*. 1992, vol. 267, n.º 1, págs. 66-72. Url: ⟨http://www.nature.com/doifinder/10.1038/scientificamerican0792-66⟩.

INFANTE, A. 2008. Teamsoft: Sistema para la gestión del trabajo en equipo en el desarrollo de proyectos de software. *Versión 2.0. Módulo de Gestión de Recursos Humanos: Trabajo para optar por el Título de Ingeniería Informática*. 2008.

ISO. 2003. *Quality Management systems - Guidelines for quality management in projects.* 2003.

J. FRANCISCO CHICANO, Enrique Alba y. 2007. Software project management with GAs. *Information Sciences*. 2007, vol. 117, n.º 11. Url: ⟨http://www.sciencedirect.com/science/article/B6V0C-4MTK976-2/2/8570bc12b346047bd32fed96dc473c3c⟩.

KELLEY JR, James E. 1961. Critical-path planning and scheduling: Mathematical basis. *Operations Research*. 1961, vol. 9, n.º 3, págs. 296-320. ISSN 1526-5463.

KIRKPATRICK, S.; GELATT, C. D. y VECCHI, M. P. 1983. Optimization by Simulated Annealing. *Science*. 1983, vol. 220, n.º 4598, págs. 671-680. Url: ⟨http://www.sciencemag.org/content/220/4598/671.abstract⟩.

KLEIN, K. 2000. Project scheduling with time-varying resource constraints. *International Journal of Production Research*. 2000, vol. 38, n.º 2, págs. 3937-3952. Url: ⟨http://dx.doi.org/DOI:10.1080/00207540050176094⟩.

KOLISCH, Rainer. 1996. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*. 1996, vol. 90, n.º 2, págs. 320-333. ISSN 0377-2217.

KOLISCH, Rainer y HARTMANN, Sonke. 1999. *Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis*. 1999.

KOLISCH, Rainer y PADMAN, Rema. 2001. An integrated survey of deterministic project scheduling. *Omega*. 2001, vol. 29, n.º 3, págs. 249-272. ISSN 0305-0483.

KOLISCH, Rainer y SPRECHER, Arno. 1997. PSPLIB - A project scheduling problem library : OR Software - ORSEP Operations Research Software Exchange Program. *European Journal of Operational Research*. 1997, vol. 96, n.º 1, págs. 205-216. Url: ⟨http://www.sciencedirect.com/science/article/B6VCT-3T7HK9P-1F/2/f3cc7f46a925673bfab16f6be5a4de4b⟩. ISSN 0377-2217.

KUMARI, A.Charan y SRINIVAS, K. 2013. Sheduling and inspection planning in software development projects using multi-objetive hyper-heuritic evolutionary algorithm. *International Journal of Software Engineering & Applications (IJSEA)*. 2013, vol. 4, n.º 3. Url: ⟨http://dx.doi.org/0.5121/ijsea.2013.4304⟩.

LAGUNA, Manuel; MARTI, Rafael y MARTÍ, Rafael Cunquero. 2003. *Scatter search: methodology and implementations in C*. 2003.

LANCASTER, J. y OZBAYRAK, M. 2007. Evolutionary algorithms applied to project scheduling problems - a survey of the state-of-the-art. *International Journal of Production Research*. 2007, vol. 45, n.º 2, págs. 425-450. ISSN 1366-588X.

MALCOLM, Donald G; ROSEBOOM, John H; CLARK, Charles E y FAZAR, Willard. 1959. Application of a technique for research and development program evaluation. *Operations research*. 1959, vol. 7, n.º 5, págs. 646-669.

MEDRANO BROCHE, Bolivar E. 2012. *Modelo para la planificación de múltiples proyectos de desarrollo de software*. 2012.

MELIÁN, Belén; PÉREZ, José A Moreno y VEGA, J Marcos Moreno. 2003. Metaheuristics: A global view. *Santa Cruz de Tenerife*. 2003.

MERKLE, Daniel; MIDDENDORF, Martin y SCHMECK, Hartmut. 2002. Ant colony optimization for resource-constrained project scheduling. *Evolutionary Computation, IEEE Transactions on*. 2002, vol. 6, n.º 4, págs. 333-346. ISSN 1089-778X.

MINISTROS, CONSEJO DE. 2014. *Decreto No. 327 Reglamento del proceso inversionista.* 2014.

MODER, Joseph John y PHILLIPS, Cecil R. 1964. *Project Management with CPM and PERT*. 1964. ISBN 0442156669.

MORILLO, Daniel; MORENO, Luis y DÍAZ, Javier. 2014. Metodologías analíticas y heurísticas para la solución del Problema de Programación de Tareas con Recursos Restringidos (RCPSP): una revisión. Parte 2. *Ingeniería y Ciencia-ing. cienc.* 2014, vol. 10, n.º 20, págs. 203-227. ISSN 1794-9165.

NARVÁEZ MOLINA Gabriela y Saltos Atiencia, Ramiro. 2010. *Implementación de un Algoritmo Genético para resolver el Problema de Programación de Proyectos con Recursos Limitados*. 2010.

PCC. 2011. *Lineamientos de la Política Económica y Social del Partido y la Revolución.* 2011. La Habana.

PÉREZ, Pedro Y. Piñero. 2010. Modelo de producción de la Universidad de las Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas*. 2010. ISSN 2227-1899.

PMI. 2013. *A Guide to the Project Management Body of Knowledge (PMBOK)*. 2013.

POSADA, Mariamar Cervantes. 2010. *Nuevos métodos Meta Heurísticos para la asignación eficiente, optimizada y robusta de recursos limitados*. 2010.

PRESSMAN, Roger S. 2005. *Software engineering: a practitioner's approach*. 2005. ISBN 0-07-285318-2.

PRINCE2. 2009. *Managing Successful Projects with Prince2*. 2009.

PRITSKER, A Alan B; WAITERS, Lawrence J y WOLFE, Philip M. 1969. Multiproject scheduling with limited resources: A zero-one programming approach. *Management science*. 1969, vol. 16, n.º 1, págs. 93-108. ISSN 1526-5501.

RAMOS, K. 2011. Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas*. 2011, vol. 5, págs. 1-16.

SANTANA, J Brito; RODRÍGUEZ, C Campos; LÓPEZ, FC García et al., 2004. Metaheurísticas: Una revisión actualizada. *Universidad de La Laguna, España: Departamento de Estadística, Investigación Operativa y Computación*. 2004, vol. 263. ISSN 1381-1231.

SEI. 2010. *CMMI for Dev. v1.3*. 2010.

STELLINGWERF, Rommert y ZANDHUIS, Anton. 2013. *ISO 21500 Guidance on project management–A Pocket Guide*. 2013.

VELASCO, Oscar Germán Duarte. 2002. *UNGenético: Una librería en C++ de Algoritmos Genéticos con Codifcación Híbrida*. 2002.

VICTORIO, Javier Navascués Fernández. 2008. *Un modelo para la simulación híbrida de la producción de software a medida en un entorno multiproyecto*. 2008.

ZOULFAGHARI, Hossein; NEMATIAN, Javad; MAHMOUDI, Nader y KHODABANDEH, Mehdi. 2013. A New Genetic Algorithm for the RCPSP in Large Scale. *International Journal of Applied Evolutionary Computation*. 2013, vol. 10, n.º 20. ISSN 1942-3594.

Generado con LaTeX: 29 de septiembre de 2015: 12:03am

Apéndices

APÉNDICE A

Resultados de Instancias de PSPLib y Reporte Generado

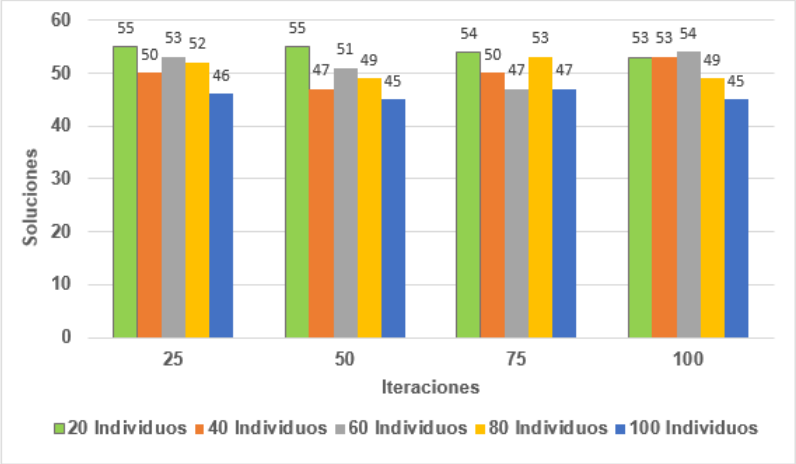


Figura A.1. Resultados para la instancia de prueba *j301-1*

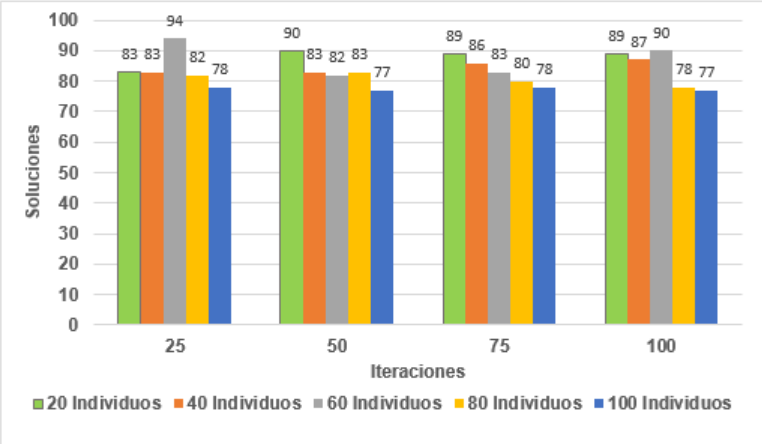


Figura A.2. Resultados para la instancia de prueba *j601-1*

Universidad de las Ciencias Informáticas

Vertex, Entornos Interactivos 3D, Facultad 5

Reporte del Algoritmo Genético para la planificación de tareas de un proyecto de software

Proyecto: Instancia de 30 tareas

Cantidad de Tareas: 32

Cantidad de Soluciones: 80

Recursos:

Tipo Recurso	Cantidad
tipo1	12
tipo2	13
tipo3	4
tipo4	12

Mejores secuencias de planificación de tareas obtenidas:

Solución 1

Tiempo Fin: 47

Núm.	Nombre Actividad	Recursos Asignados
1	Actividad 1	*Actividad Ficticia*
4	Actividad 4	Lorgio, Kit, Fernando
9	Actividad 9	Esmaykel, Miguel, Pepe, Lolo, Pier, Tai
3	Actividad 3	Jorge, Leonardo, Mary, Cristina, Joy, k, Esmaykel, Miguel, Pepe, Lolo
8	Actividad 8	Mario
12	Actividad 12	Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
13	Actividad 13	Jorge, Leonardo, Mary, Cristina
5	Actividad 5	Esmaykel, Miguel, Pepe
7	Actividad 7	Lolo, Joy, k, Pier
18	Actividad 18	Fernanda, Joaquin, Maykel, Gabriel, Gabriele, Susej, Ana
14	Actividad 14	Julia, Yoe, Yane, Frank, Yuly, Mario, Enrique, Luis
10	Actividad 10	Anatol
16	Actividad 16	Angel, Lorgio, Kit, Fernando, Fernanda
21	Actividad 21	Joaquin, Maykel, Gabriel, Gabriele, Susej, Ana
19	Actividad 19	Pit
2	Actividad 2	Tai, Jorge, Leonardo, Mary
11	Actividad 11	Lucas, Walter, Migue, Julio, Mario
15	Actividad 15	Cristina, Esmaykel, Miguel
17	Actividad 17	Anatol, Lorgio, Kit, Fernando, Fernanda, Angel, Joaquin, Maykel
6	Actividad 6	Gabriel, Gabriele, Susej, Ana, Lorgio, Kit, Fernando, Fernanda
27	Actividad 27	Joaquin, Maykel, Anatol, Gabriel, Gabriele, Susej, Ana
29	Actividad 29	Enrique, Luis, Pit, Julia, Yoe, Yane, Frank
20	Actividad 20	Yuly, Enrique, Luis, Pit, Julia, Yoe, Yane, Frank, Mario, Lucas
25	Actividad 25	Pepe, Lolo, Pier, Joy
26	Actividad 26	Josep, Josepe, Martha, Julita
28	Actividad 28	Enrique, Luis, Pit, Julia, Yoe, Yane, Frank, Yuly
22	Actividad 22	k, Tai
23	Actividad 23	Jorge, Leonardo, Mary
24	Actividad 24	Enrique, Luis, Pit, Julia, Yoe, Yane, Frank, Yuly, Walter
31	Actividad 31	Josep, Josepe

30	Actividad 30	Migue, Julio, Mario, Lucas, Enrique, Luis, Pit
32	Actividad 32	*Actividad Ficticia*

Solución 2

Tiempo Fin: 47

Núm.	Nombre Actividad	Recursos Asignados
1	Actividad 1	*Actividad Ficticia*
4	Actividad 4	Lorgio, Kit, Fernando
9	Actividad 9	Esmaykel, Miguel, Pepe, Lolo, Pier, Tai
3	Actividad 3	Jorge, Leonardo, Mary, Cristina, Joy, k, Esmaykel, Miguel, Pepe, Lolo
8	Actividad 8	Mario
12	Actividad 12	Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
13	Actividad 13	Jorge, Leonardo, Mary, Cristina
5	Actividad 5	Fernanda
7	Actividad 7	Julia, Yoe, Yane, Frank, Yuly
18	Actividad 18	Josep, Josepe, Martha, Julita
14	Actividad 14	Joaquin, Maykel, Gabriel, Gabriele, Susej, Ana, Anatol, Angel
10	Actividad 10	Esmaykel, Miguel, Pepe, Lolo
16	Actividad 16	Joy, k, Pier
21	Actividad 21	Tai, Pier, Joy
19	Actividad 19	Tai, Jorge, Leonardo, Mary
2	Actividad 2	Lorgio, Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel
11	Actividad 11	Mario
15	Actividad 15	Gabriele, Susej, Ana, Anatol, Angel
17	Actividad 17	Lorgio, Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel
6	Actividad 6	Gabriele, Susej, Ana, Anatol, Angel, Lorgio
27	Actividad 27	Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
29	Actividad 29	Mario, Julia, Yoe, Yane, Frank, Yuly, Enrique, Luis, Pit, Lucas
20	Actividad 20	Walter, Migue, Julio, Mario, Enrique, Luis, Pit, Lucas
25	Actividad 25	Mario, Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
26	Actividad 26	Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel, Lorgio, Gabriele
22	Actividad 22	Pier, Joy, Cristina, Esmaykel
28	Actividad 28	Miguel, Pepe
31	Actividad 31	Lolo, Tai, Jorge
23	Actividad 23	Julia, Yoe, Yane, Frank, Yuly, Mario, Enrique, Luis, Pit
24	Actividad 24	Josep, Josepe
30	Actividad 30	Lucas, Walter, Migue, Julio, Mario, Enrique, Luis
32	Actividad 32	*Actividad Ficticia*

Solución 3

Tiempo Fin: 54

Núm.	Nombre Actividad	Recursos Asignados
1	Actividad 1	*Actividad Ficticia*
4	Actividad 4	Lorgio, Kit, Fernando
9	Actividad 9	Esmaykel, Miguel, Pepe, Lolo, Pier, Tai
3	Actividad 3	Jorge, Leonardo, Mary, Cristina, Joy, k, Esmaykel, Miguel, Pepe, Lolo
8	Actividad 8	Mario
12	Actividad 12	Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
2	Actividad 2	Jorge, Leonardo, Mary, Cristina
10	Actividad 10	Fernanda
11	Actividad 11	Julia, Yoe, Yane, Frank, Yuly
26	Actividad 26	Josep, Josepe, Martha, Julita
6	Actividad 6	Joaquin, Maykel, Gabriel, Gabriele, Susej, Ana, Anatol, Angel
7	Actividad 7	Esmaykel, Miguel, Pepe, Lolo
15	Actividad 15	Joy, k, Pier
5	Actividad 5	Tai, Pier, Joy
13	Actividad 13	Tai, Jorge, Leonardo, Mary
16	Actividad 16	Lorgio, Kit, Fernando, Fernanda, Joaquin
21	Actividad 21	Maykel, Gabriel, Gabriele, Susej, Ana, Anatol
18	Actividad 18	Angel, Maykel, Gabriel, Gabriele, Susej, Ana, Anatol

20	Actividad 20	Mario, Enrique, Luis, Pit, Lucas, Walter, Migue, Julio, Julia, Yoe
19	Actividad 19	Mario
14	Actividad 14	Mario, Enrique, Luis, Pit, Lucas, Walter, Migue, Julio
17	Actividad 17	Maykel, Gabriel, Gabriele, Susej, Ana, Anatol, Angel, Lorgio
29	Actividad 29	Mario, Enrique, Luis, Pit, Lucas, Walter, Migue
27	Actividad 27	Kit, Fernando, Fernanda, Joaquin, Lorgio, Maykel, Gabriel
22	Actividad 22	Pier, Joy
28	Actividad 28	Julio, Yane, Frank, Yuly, Julia, Yoe, Mario, Enrique
25	Actividad 25	Cristina, Esmaykel, Miguel, Pepe
23	Actividad 23	Lolo, Tai, Jorge
31	Actividad 31	Josep, Josepe
24	Actividad 24	Luis, Pit, Lucas, Walter, Migue, Mario, Enrique, Julio, Julia
30	Actividad 30	Yoe, Yane, Frank, Yuly, Mario, Enrique, Luis
32	Actividad 32	*Actividad Ficticia*

Solución 4

Tiempo Fin: 54

Núm.	Nombre Actividad	Recursos Asignados
1	Actividad 1	*Actividad Ficticia*
3	Actividad 3	Esmaykel, Miguel, Pepe, Lolo, Pier, Tai, Jorge, Leonardo, Mary, Cristina
2	Actividad 2	Joy, k, Esmaykel, Miguel
6	Actividad 6	Lorgio, Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel, Gabriele
4	Actividad 4	Susej, Ana, Anatol
9	Actividad 9	Pepe, Lolo, Pier, Tai, Jorge, Leonardo
8	Actividad 8	Mario
7	Actividad 7	Mary, Cristina, Pepe, Lolo
15	Actividad 15	Pepe, Lolo, Mary
5	Actividad 5	Cristina, Esmaykel, Miguel
13	Actividad 13	Pier, Tai, Jorge, Leonardo
11	Actividad 11	Enrique, Luis, Pit, Lucas, Walter
10	Actividad 10	Angel
16	Actividad 16	Susej, Ana, Anatol, Angel, Lorgio
21	Actividad 21	Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel
12	Actividad 12	Migue, Julio, Julia, Yoe, Yane, Frank, Yuly
26	Actividad 26	Josep, Josepe, Martha, Julita
18	Actividad 18	Gabriele, Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel
20	Actividad 20	Mario, Migue, Julio, Julia, Yoe, Yane, Frank, Yuly, Enrique, Luis
19	Actividad 19	Mario
14	Actividad 14	Mario, Migue, Julio, Julia, Yoe, Yane, Frank, Yuly
17	Actividad 17	Kit, Fernando, Fernanda, Joaquin, Maykel, Gabriel, Gabriele, Lorgio
29	Actividad 29	Mario, Migue, Julio, Julia, Yoe, Yane, Frank
27	Actividad 27	Susej, Ana, Anatol, Angel, Lorgio, Kit, Fernando
22	Actividad 22	Pier, Tai
28	Actividad 28	Yuly, Pit, Lucas, Walter, Enrique, Luis, Mario, Migue
25	Actividad 25	Jorge, Leonardo, Joy, k
31	Actividad 31	Josep, Josepe
23	Actividad 23	Esmaykel, Miguel, Cristina
24	Actividad 24	Julio, Julia, Yoe, Yane, Frank, Mario, Enrique, Luis, Pit
30	Actividad 30	Lucas, Walter, Migue, Yuly, Mario, Enrique, Luis
32	Actividad 32	*Actividad Ficticia*

Solución 5

Tiempo Fin: 54

Núm.	Nombre Actividad	Recursos Asignados
1	Actividad 1	*Actividad Ficticia*
4	Actividad 4	Lorgio, Kit, Fernando
9	Actividad 9	Esmaykel, Miguel, Pepe, Lolo, Pier, Tai
3	Actividad 3	Jorge, Leonardo, Mary, Cristina, Joy, k, Esmaykel, Miguel, Pepe, Lolo
8	Actividad 8	Mario
2	Actividad 2	Jorge, Leonardo, Mary, Cristina

6	Actividad 6	Fernanda, Joaquin, Maykel, Gabriel, Gabriele, Susej, Ana, Anatol
11	Actividad 11	Enrique, Luis, Pit, Lucas, Walter
15	Actividad 15	Esmaykel, Miguel, Pepe
5	Actividad 5	Lolo, Joy, k
7	Actividad 7	Pier, Tai, Lolo, Joy
10	Actividad 10	Angel
12	Actividad 12	Migue, Julio, Julia, Yoe, Yane, Frank, Yuly
13	Actividad 13	k, Jorge, Leonardo, Mary
18	Actividad 18	Lorgio, Kit, Fernando, Angel, Fernanda, Joaquin, Maykel
14	Actividad 14	Mario, Migue, Julio, Julia, Yoe, Yane, Frank, Yuly
16	Actividad 16	Gabriel, Gabriele, Susej, Ana, Anatol
21	Actividad 21	Lorgio, Kit, Fernando, Fernanda, Joaquin, Maykel
26	Actividad 26	Josep, Josepe, Martha, Julita
20	Actividad 20	Mario, Migue, Julio, Julia, Yoe, Yane, Frank, Yuly, Enrique, Luis
19	Actividad 19	Mario
17	Actividad 17	Angel, Gabriel, Gabriele, Susej, Ana, Anatol, Lorgio, Kit
29	Actividad 29	Pit, Lucas, Walter, Mario, Enrique, Luis, Migue
27	Actividad 27	Fernando, Fernanda, Joaquin, Maykel, Lorgio, Kit, Gabriel
28	Actividad 28	Julio, Julia, Yoe, Yane, Frank, Yuly, Mario, Enrique
25	Actividad 25	Cristina, Lolo, Pier, Tai
31	Actividad 31	Josep, Josepe
22	Actividad 22	Joy, Jorge
23	Actividad 23	Leonardo, Mary, k
24	Actividad 24	Luis, Pit, Lucas, Walter, Migue, Mario, Enrique, Julio, Julia
30	Actividad 30	Yoe, Yane, Frank, Yuly, Mario, Enrique, Luis
32	Actividad 32	*Actividad Ficticia*

Este reporte fue generado en Qt Creator el 02/06/2015