

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**DISEÑO DE UN ALGORITMO DE BÚSQUEDA TABÚ PARA
RESOLVER EL PROBLEMA DE LA SELECCIÓN DE
PROYECTOS**

Tesis para optar el Título de Ingeniero Informático que presenta el bachiller:

Eduardo Antonio Rejas Cano

ASESOR: Ing. Rony Cueva

Lima, Julio de 2014

RESUMEN

La Selección de Proyectos de Tecnología de Información es importante en la actualidad ya que gracias a estos se consiguen ventajas competitivas que permiten a la empresa en cuestión marcar diferencia en el mercado y generar ventaja competitiva. Por ello una solución que otorgue utilidades y satisfaga las expectativas de la gerencia es indispensable, es por esta razón que se propone un algoritmo metaheurístico que cumpla con dichos requisitos.

La propuesta es la implementación de un algoritmo de Búsqueda Tabú (Tabu Search) de tres fases (Básica, Intensificación y Diversificación) que optimice las utilidades de un portafolio de proyectos de Tecnologías de Información. Un punto importante a tener en cuenta es que este algoritmo llega a la solución en un menor tiempo que otros métodos existentes, como son los modelos matemáticos y de simulación, obteniendo resultados iguales o mejores que con los métodos mencionados. Para tener la certeza de que la solución obtenida es buena, se contrastó con otro algoritmo de relativa complejidad (GRASP construcción) mediante métodos estadísticos, teniendo como resultado que la media del algoritmo de Búsqueda Tabú es mayor y por tanto mejor que la del GRASP.

Finalmente, se demuestra que la solución propuesta, un algoritmo de Búsqueda Tabú para la selección de proyectos de Tecnología de Información, es una opción a tomar en cuenta para la toma de decisiones al momento de armar un portafolio de proyectos que permita a la empresa generar utilidades y ventaja competitiva.

Tabla de contenido

CAPÍTULO 1	7
1. PROBLEMÁTICA.....	7
2. MARCO TEÓRICO	9
2.1 <i>Marco conceptual</i>	9
2.1.1 Optimización Combinatoria.....	9
2.1.2 Problema de la Mochila (Knapsack Problem)	11
2.1.3 Conceptos generales de heurísticas y meta-heurísticas	11
3. ESTADO DEL ARTE.....	20
3.1 <i>Formas exactas de resolver el problema</i>	20
3.1.1 Multi-Criteria Decision Making (MCDM)	20
3.1.2 Information Techonology Strategic Research (ITSR) Project.....	20
3.1.3 Algoritmo genético para resolver el problema de selección de proyectos	21
3.2 <i>Formas aproximadas de resolver el problema</i>	22
3.2.1 The Art of Project Selection	22
3.2.2 Algoritmo GRASP para resolver el problema de selección de proyectos	23
3.2.3 A Multiple Criteria 0-1 Linear Programming Method of Project Planning	
and Investment Decision Making.....	24
3.2.4 An Immune Algorithm for Optional Selection Problem of Investment	
Projects.....	25
3.3 <i>Productos comerciales para resolver el problema</i>	26
3.3.1 Augeo Software	26
3.3.2 Project Management Office	26
3.3.3 Project Portfolio Optimization.....	27
3.4 <i>Productos de investigación para resolver el problema</i>	28
3.4.1 A Research on Multiple Projects Selection Base on Matching Matrix	
Model	28
3.4.2 The research of the financial Evaluation Methods of investment Projects	
Based on EVA.....	29
3.5 <i>Conclusiones sobre el estado del arte</i>	29
CAPÍTULO 2	30
1. OBJETIVO GENERAL	30
2. OBJETIVOS ESPECÍFICOS.....	30

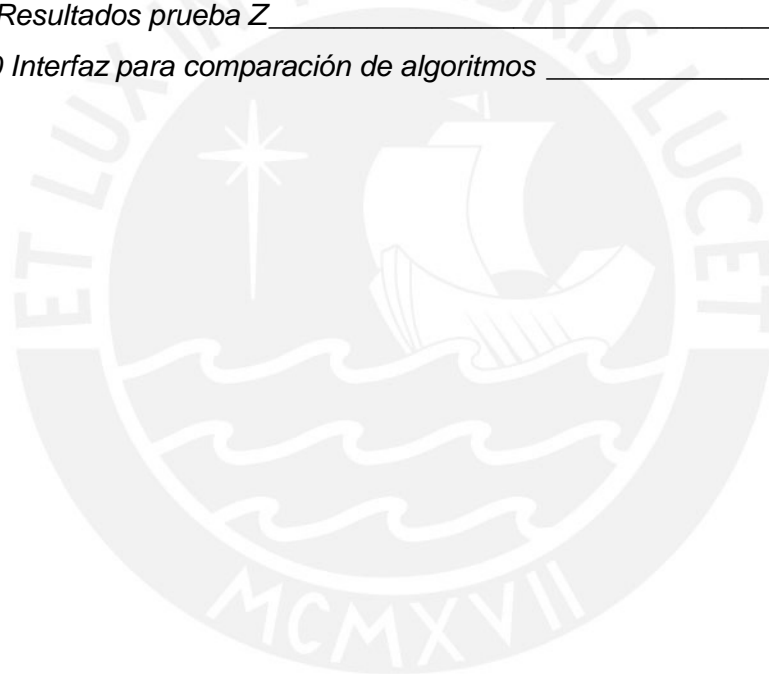
3.	RESULTADOS ESPERADOS	31
4.	HERRAMIENTAS, MÉTODOS Y PROCEDIMIENTOS	32
4.1	<i>Mapeo</i>	32
4.2	<i>Herramientas para el desarrollo</i>	34
4.3	<i>Herramientas para la gestión</i>	34
4.4	<i>Herramientas Metodológicas</i>	40
4.5	<i>Otras herramientas</i>	42
5.	ALCANCE.....	43
5.1	<i>Limitaciones</i>	44
5.2	<i>Riesgos</i>	45
6.	JUSTIFICACIÓN Y VIABILIDAD	46
6.1	<i>Justificativa del proyecto de tesis</i>	46
6.2	<i>Análisis de viabilidad del proyecto de tesis</i>	47
CAPÍTULO 3	49
1.	HERRAMIENTA GENERADORA DE DATOS.....	49
CAPÍTULO 4	51
1.	ESTRUCTURAS DE DATOS	51
CAPÍTULO 5	55
1.	FUNCIÓN OBJETIVO	55
CAPÍTULO 6	57
1.	PSEUDOCÓDIGO DE ALGORITMO GRASP CONSTRUCCIÓN.....	57
CAPÍTULO 7	61
1.	PSEUDOCÓDIGO DE ALGORITMO BÚSQUEDA TABÚ	61
CAPÍTULO 8	69
1.	EXPERIMENTACIÓN NUMÉRICA.....	69
1.1	<i>Prueba Kolmogorov Smirnov</i>	70
1.2	<i>Prueba F de Fischer</i>	72
1.3	<i>Prueba Z</i>	73
CAPÍTULO 9	76
1.	INTERFASE	76
CAPÍTULO 10	78
1.	CONCLUSIONES	78

2. RECOMENDACIONES Y TRABAJOS FUTUROS	79
REFERENCIAS BIBLIOGRÁFICAS.....	80



ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1 Cuatro Dimensiones de la Búsqueda Tabú</i>	17
<i>Ilustración 2 Reporte de Project Management Office</i>	27
<i>Ilustración 3 Reporte de Project Portfolio Optimization</i>	28
<i>Ilustración 4 Pantalla de herramienta generadora de datos</i>	49
<i>Ilustración 5 Muestra de datos obtenidos</i>	70
<i>Ilustración 6 Resultado Kolmogorov-Smirnov (GRASP construcción)</i>	71
<i>Ilustración 7 Resultado prueba Kolmogorov-Smirnov (Búsqueda Tabú)</i>	71
<i>Ilustración 8 Resultados prueba F de Fischer</i>	73
<i>Ilustración 9 Resultados prueba Z</i>	74
<i>Ilustración 10 Interfaz para comparación de algoritmos</i>	76



ÍNDICE DE TABLAS

<i>Tabla 1 Procedimiento GRASP</i>	14
<i>Tabla 2 Búsqueda por entorno</i>	18
<i>Tabla 3 Algoritmo Genético para la selección de proyectos</i>	22
<i>Tabla 4 Mapeo de Herramientas</i>	33
<i>Tabla 5 Herramientas para la gestión</i>	39
<i>Tabla 6 Riesgos del proyecto</i>	45
<i>Tabla 7 Estructura herramienta generadora de datos</i>	50
<i>Tabla 8 Variables del problema</i>	51
<i>Tabla 9 Estructuras de Datos (RCL)</i>	51
<i>Tabla 10 Estructuras de Datos (Matriz de proyectos)</i>	52
<i>Tabla 11: Tabla de estados de riesgo (R)</i>	52
<i>Tabla 12 Estructuras de datos (Matriz de dependencia de proyectos)</i>	52
<i>Tabla 13 Estructuras de datos (Memoria de lo reciente)</i>	53
<i>Tabla 14 Estructuras de datos (Memoria de lo frecuente)</i>	54
<i>Tabla 15 Variables para algoritmo GRASP construcción</i>	57
<i>Tabla 16 Estructuras de datos para algoritmo GRASP construcción</i>	57
<i>Tabla 17 Pseudocódigo GRASP construcción</i>	58
<i>Tabla 18 Variables para algoritmo Búsqueda Tabú</i>	61
<i>Tabla 19 Estructuras de datos para algoritmo Búsqueda Tabú</i>	61
<i>Tabla 20 Pseudocódigo Búsqueda Tabú</i>	62
<i>Tabla 21 Procedimiento Búsqueda (Búsqueda Tabú)</i>	63
<i>Tabla 22 Procedimiento Identificar Vecinos (Búsqueda Tabú)</i>	63
<i>Tabla 23 Procedimiento Selecciona Mejor Movimiento (Búsqueda Tabú)</i>	64

CAPÍTULO 1

1. Problemática

En la actualidad, las empresas tienen la necesidad de invertir en proyectos de modo que estos generen ganancia y crecimiento. En un inicio, se seleccionaban aquellos proyectos que, según intuición, eran los que más ganancia producían. Posteriormente, se empezaron a elaborar modelos matemáticos que permitían evaluar uno a uno los proyectos, para luego seleccionarlos. Sin embargo, en la actualidad, es más conveniente evaluar los proyectos en grupos o portafolios, de modo que estos puedan resolver las necesidades de la organización [CARAZO, GÓMEZ, PÉREZ, 2011].

Uno de los tipos de proyectos más importantes a elegir son los proyectos de tecnologías de información (TI), ya que estos permitirán tomar ventaja competitiva sobre otras empresas, siempre y cuando se haga una buena selección de proyectos.

Para la adecuada selección de proyectos, en este caso de TI, se debe tener en cuenta que hay factores que influyen en la toma de decisiones, estos factores son llamados agentes decisores, los cuales determinan la selección del proyecto. Estos factores a tener en cuenta son la interacción o dependencia de proyectos, el tiempo estimado por cada proyecto, el análisis del riesgo y finalmente el que sea multiobjetivo, es decir, que cumpla varios objetivos a la vez [CARAZO, GÓMEZ, PÉREZ, 2011]. Adicionalmente a los factores mencionados, se puede agregar el costo y la prioridad de cada uno de ellos.

Para empezar, se tiene que la dependencia de proyectos puede volver engorrosa la elección de una posible alternativa de solución, aun cuando se esté evaluando un pequeño conjunto de proyectos con pocos objetivos de optimización [DICKINSON, THORNTON, GRAVES, 2001]. Esta dependencia se observa al momento de elegir entre las distintas opciones, ya que en algunos casos se requiere de un proyecto para ejecutar otro (que podría ser más de uno). Muchas veces esto no es posible puesto que se puede exceder el presupuesto asignado o sencillamente uno de los proyectos precedentes no es rentable para la organización lo que haría inviable esa alternativa.

El siguiente factor es el tiempo, que es un estimado de la duración de cada proyecto, el cual no debe sobrepasar el horizonte temporal en el que se desarrollará la inversión del portafolio. Este factor puede estar ligado a la dependencia de proyectos.

Otros factores a considerar en la selección de proyectos son los riesgos, los cuales cada proyecto tiene de forma inherente. Dichos riesgos son las probabilidades de que se pueda producir pérdidas si no se evalúan y controlan adecuadamente. Sin embargo, en la mayoría de casos, se cumple que a mayor riesgo, mayor será la ganancia [CHEN DONG, 2010]. Por lo referido anteriormente, se deduce que la complejidad para seleccionar aquellos proyectos que garanticen ganancia, aumentan si existen muchos proyectos a evaluar. Por ello, se debe tener mucho cuidado al elegir los proyectos que deben ir en un portafolio, pues en algunos casos esos riesgos pueden desencadenar en pérdidas.

El último factor a considerar es que el proyecto o los proyectos seleccionados cumplan, en la medida de lo posible, más de un objetivo a la vez. Un caso muy común es el de minimizar los riesgos y maximizar ganancias [FULGA, DEDU, 2010]. Se podría minimizar, del mismo modo, los costos de inversión.

Todos los factores expuestos anteriormente se vuelven más complejos de lo que son cuando tratamos con grandes cantidades de proyectos a elegir. Ya que llegar a una solución exacta sería muy complicado y tomaría demasiado tiempo el evaluar cada proyecto en base a cada factor, este proyecto de fin de carrera pretende solucionar este problema mediante el uso de un algoritmo metaheurístico. Este algoritmo no resolverá el problema de forma exacta debido a la complejidad del mismo, sin embargo se acercará de manera aceptable a la solución óptima, cuyo principal objetivo es el de optimizar costo beneficio. De este modo las empresas que inviertan en proyectos de TI ahorrarán tiempo y dinero, lo que les permitirá enfocarse en otras áreas del negocio.

2. Marco teórico

2.1 Marco conceptual

A continuación se presentarán algunos conceptos que permitirán conocer mejor el problema, así como la solución a plantear. Entre estos se tiene la optimización combinatoria, el problema de la mochila (Knapsack Problem), conceptos generales de heurísticas y meta-heurísticas, y algunos algoritmos.

2.1.1 Optimización Combinatoria

Cada problema existente requiere una solución, y con frecuencia se pretende que sea óptima. Por ejemplo:

- Idear un plan de ventas que permita maximizar ganancias.
- Encontrar una distribución adecuada de productos para minimizar el espacio que ocupan.
- Recorrer diferentes lugares de una ciudad minimizando la distancia recorrida.

Estos problemas tratan de minimizar o maximizar ciertos valores por lo que se les llama problemas de optimización (MORENO, HUECAS, SÁNCHEZ, 2007). Sin embargo, para algunos de estos problemas no es posible encontrar una solución exacta, ya los universos de datos son demasiado grandes y sería imposible terminar de analizarlos.

Los problemas de optimización pueden ser divididos en dos categorías:

- Con variables continuas: Se espera obtener un conjunto de números.

- Con variables discretas: Llamados combinatorios. Se espera obtener un objeto, que puede ser un número o un conjunto de números a partir de un universo finito de datos [PAPADIMITRIOU, STEIGLITZ, 1998]. En estos casos, es preferible tener una solución aproximada y que esté en un rango donde se considere que la solución propuesta es aceptable.

Ante esta limitación, la optimización combinatoria pretende dar solución a estos problemas bajo la premisa de “plantear métodos algorítmicos que permiten realizar combinaciones y ordenamientos a grupos de elementos que forman parte de determinados universos (concepto que se puede asociar con el de espacio de estados) con el fin de conseguir patrones adecuados que resuelvan determinados tipos de problema considerados por la misma algorítmica como de complejidad NP o NP-difícil [TUPIA, 2009].”

De modo general, un problema de optimización combinatoria puede ser expresado como [TUPIA, 2009]:

Minimizar: $f(x)$

Sujeto a: $g_i(x) \geq 0 \quad i = 1, \dots, m$

$h_j(x) \geq 0 \quad j = 1 \dots n$

Donde:

$f(x)$ = función objetivo que representa un valor a optimizar.

$g_i(x) \geq 0 \quad i = 1, \dots, m$ = restricción 1

$h_j(x) \geq 0 \quad j = 1 \dots n$ = restricción 2

2.1.2 Problema de la Mochila (Knapsack Problem)

El problema de la mochila o Knapsack Problem consiste tener una mochila de cierta capacidad, que debe ser llenada con objetos de forma tal que se obtenga el mayor beneficio posible. Este puede ser representado matemáticamente de la siguiente manera [MARTELLO, TOTH, 1990]:

$$\text{Maximizar: } \sum_{j=1}^n p_j x_j$$

$$\text{Sujeto a: } \sum_{j=1}^n w_j x_j \leq c$$

Donde:

$x = 1$ si objeto j es seleccionado, 0 si no es seleccionado.

p = beneficio esperado.

c = capacidad de la mochila.

n = número de objetos de entre los cuales se elegirá.

w = peso o tamaño del objeto.

Bajo este concepto, se puede adaptar este mismo problema al de la selección y organización de proyectos, donde se cuenta con un presupuesto y un número de proyectos candidatos. En este caso, se deberá seleccionar aquellos proyectos que generen mayor beneficio. Adicionalmente, se podría agregar otros factores como riesgo y tiempo del proyecto.

2.1.3 Conceptos generales de heurísticas y meta-heurísticas

Para resolver los problemas de optimización, las técnicas heurísticas y meta heurísticas son una buena elección. Si bien son métodos aproximados, ofrecen soluciones que son buenas y se encuentran dentro rango de aceptación ya que, como se mencionó en el apartado de optimización combinatoria, muchas veces no es posible hallar la solución exacta.

2.1.3.1 Algoritmos Heurísticos

Son procedimientos que exploran un conjunto de soluciones o candidatos, evaluándolos de manera progresiva y dirigiendo la búsqueda hacia un resultado que no es del todo óptimo, ya que no analiza todas las posibles combinaciones [TUPIA, 2009].

Los algoritmos heurísticos pueden clasificarse en [MARTI, 2007]:

a) **Métodos Constructivos:** Permiten construir paso a paso la solución a un problema dado. Estos pasos a seguir dependen la estrategia que se vaya a seguir. Entre las más comunes se tienen:

- **Estrategia voraz:** Partiendo del problema se va construyendo una solución factible. Consiste en obtener soluciones parciales que se caracterizan por ser las mejores en ese momento. Sin embargo, esto es una desventaja ya que eligen la mejor alternativa actual pero esto puede pesar en el futuro, al obtener la solución completa. Por este motivo se dice que son “miopes”.
- **Estrategia de descomposición:** Consiste en dividir el problema en otros más pequeños (divide y vencerás). Esta división continúa hasta que la solución a esos pequeños subproblemas sea bastante sencilla.
- **Método de reducción:** Se van identificando características que tienen las mejores soluciones halladas. En base a este procedimiento, se asume que la mejor solución también poseerá estas mismas características, lo que hace que el espacio de búsqueda de candidatos se vea simplificado.

b) **Métodos de Búsqueda:** Se trata de mejorar una solución factible hallada. Se tienen las siguientes estrategias:

- **Estrategia de búsqueda local 1:** Dada una solución factible, trata de mejorarla de manera progresiva. Para lograrlo examina el universo de soluciones donde se encuentra y selecciona la primera que produzca una mejora (*first improvement*).

- Estrategia de búsqueda local 2: Dada una solución factible, trata de mejorarla de manera progresiva. Para lograrlo examina el universo de soluciones donde se encuentra y selecciona el mejor movimiento de todos los posibles, es decir, aquel que produzca una mayor mejora (*best improvement*).
- Estrategia aleatorizada: Dada una solución factible asociada un universo de soluciones, se selecciona de manera aleatoria una de las soluciones dentro de dicho universo.

2.1.3.2 Algoritmos Meta heurísticos

Si bien los algoritmos heurísticos son de fácil implementación, también es conocido que tienen ciertos defectos, ya que no siempre se obtendrán buenas soluciones. Estas características defectuosas son la voracidad y miopía [TUPIA, 2009] [GALLART, 2009].

- Voracidad: siempre se elige el mejor candidato para formar parte de la solución, este candidato es aquel que tenga un mejor resultado al evaluarse la función objetivo.
- Miopía: habiéndose seleccionado un candidato para formar parte de la solución, esta elección ya no puede ser modificada y no se analizó los efectos que puede causar a futuro a la solución final. Otra definición aceptada es la que propone Rafael Martí. “Son procedimientos de alto nivel que guían a algoritmos heurísticos conocidos evitando que estos caigan en óptimos locales [MARTÍ, 2007]”.

2.1.3.3 Algoritmo GRASP (Greedy Randomize Adaptive Search Procedure)

Es un algoritmo que emplea una técnica iterativa de muestreo aleatorio, y en cada una de sus iteraciones provee una solución en la que la última obtenida es la que se toma como resultado final [FEO, RESENDE, 1995].

El algoritmo GRASP como sus siglas indican significa que es un procedimiento voraz, aleatorio, adaptativo y de búsqueda. Se dice que es voraz porque cada vez que evalúa selecciona los mejores candidatos en base a ciertas condiciones, se dice que es aleatorio porque se seleccionan al azar aquellas que formarán parte del conjunto solución relajando el criterio voraz, se dice que es adaptativo porque se puede adaptar a la estructura de la instancia de algún problema que se quiera resolver y se dice que es de búsqueda porque dentro de un universo de posibles soluciones se ejecutan búsquedas sin evaluar todos los elementos del problema.

A continuación se presenta la estructura general del algoritmo GRASP [TUPIA, 2009]:

<p>Procedimiento GRASP (Instancia del problema)</p> <ol style="list-style-type: none"> 1. Leer (Instancia) 2. Mientras «no se cumpla condición de parada» hacer <ol style="list-style-type: none"> 2.1 Procedimiento Construcción (S_1) 2.2 Procedimiento Mejorar (S_1) Fin mientras 3. Retornar (Mejor S_1) <p>Fin GRASP</p>
--

Tabla 1 Procedimiento GRASP

Este algoritmo presenta dos fases: construcción y búsqueda local o mejora.

a) Fase de construcción

En esta fase se irá construyendo la solución al problema tomando un elemento a la vez. Cada elemento será seleccionado aleatoriamente de una lista de candidatos (RCL) determinada por una función voraz [FEO, RESENDE, 1995]. A continuación un esquema de cómo armar la lista de candidatos, teniendo en cuenta que el criterio voraz que busca optimizar la función objetivo puede ser de la forma: Mejor $\{c(x) : x \in N\}$ [TUPIA, 2009]:

- Se obtienen los mejores y peores valores de la función objetivo, en este caso c para así formar la RCL :

$$\beta = \text{Mejor} \{c(x) : x \in N\}$$

$$\tau = \text{Peor} \{c(x) : x \in N\}$$

- Se forma la RCL de la siguiente forma:

$$RCL = \{x \in N : \beta \leq C(x) \leq \beta + \alpha (\tau - \beta)\}$$

Donde α es la constante de relajación y es la encargada de que la solución sea menos voraz.

Se debe tener en cuenta que si α es igual a 0, se tomará un criterio totalmente aleatorio, de igual modo si α es igual a 1, se tomara un criterio totalmente voraz. Por lo que la constante de relajación α debe ser un número entre 0 y 1, que será determinado mediante experimentación numérica.

b) Fase de búsqueda local o mejora

Como se sabe, las soluciones que se obtienen del algoritmo GRASP no necesariamente son las mejores. Por ello, esta fase pretende hacer una optimización de las soluciones ya encontradas en la fase de construcción [FEO, RESENDE, 1995]. Para lograrlo, se debe definir en primer lugar un universo de soluciones en torno de la solución hallada. Una forma de construir este universo es ir alternando uno por uno los elementos que pertenecen a la solución con los que no pertenecen a la solución, siempre y cuando estos también pertenezcan al universo de datos en torno de la solución [TUPIA, 2009].

Es importante conocer los aspectos generales del algoritmo GRASP descrito, ya que se pretende hacer una comparación entre este, y el algoritmo propuesto (Búsqueda Tabú), que se describirá en la siguiente sección. Esta comparación permitirá conocer cuál de estos dos algoritmos es el que mejor se adapta al problema planteado.

2.1.3.4 Algoritmo Búsqueda Tabú (Tabu Search)

Búsqueda Tabú es un procedimiento adaptativo de mejora que puede hacer uso de otras metodologías como la programación lineal y otras heurísticas. Tiene como objetivo superar las limitaciones del óptimo local y encontrar un óptimo global por lo que, incluso podría ser utilizado después de la fase de construcción del GRASP, en la etapa de búsqueda local o mejora [GLOVER, 1989].

Este algoritmo consiste en explotar ciertas estrategias inteligentes para resolver problemas, las cuales están basadas en procedimientos de aprendizaje, ya sean explícitos o implícitos. Esto es posible a que el algoritmo Búsqueda Tabú cuenta con una característica especial, la memoria adaptativa. El uso de esta memoria permite explotar la historia del proceso de resolución del problema; sin embargo, su dificultad radica en la necesidad de crear estructuras de memoria para que esta pueda ser explotada [GLOVER, MELIÁN, 2003].

A continuación se explicarán algunos conceptos que permitirán una mejor comprensión del algoritmo:

2.1.3.4.1 Uso de memoria

Se debe tener en cuenta que las estructuras de memoria usadas en Búsqueda Tabú, trabajan haciendo referencia a cuatro dimensiones principales, las cuales son el ser reciente, la frecuencia, la calidad y la influencia. El que sea reciente y la frecuencia se complementan la una a la otra, de modo que se hace un seguimiento a las características que han cambiado recientemente y la cantidad de veces que sucedido. La calidad consiste en la habilidad de distinguir buenas soluciones de entre las que fueron evaluadas durante la búsqueda. Finalmente, la influencia hace referencia al impacto que puede afectar la calidad de la solución durante el proceso de búsqueda de la misma [GLOVER, LAGUNA, 1997].

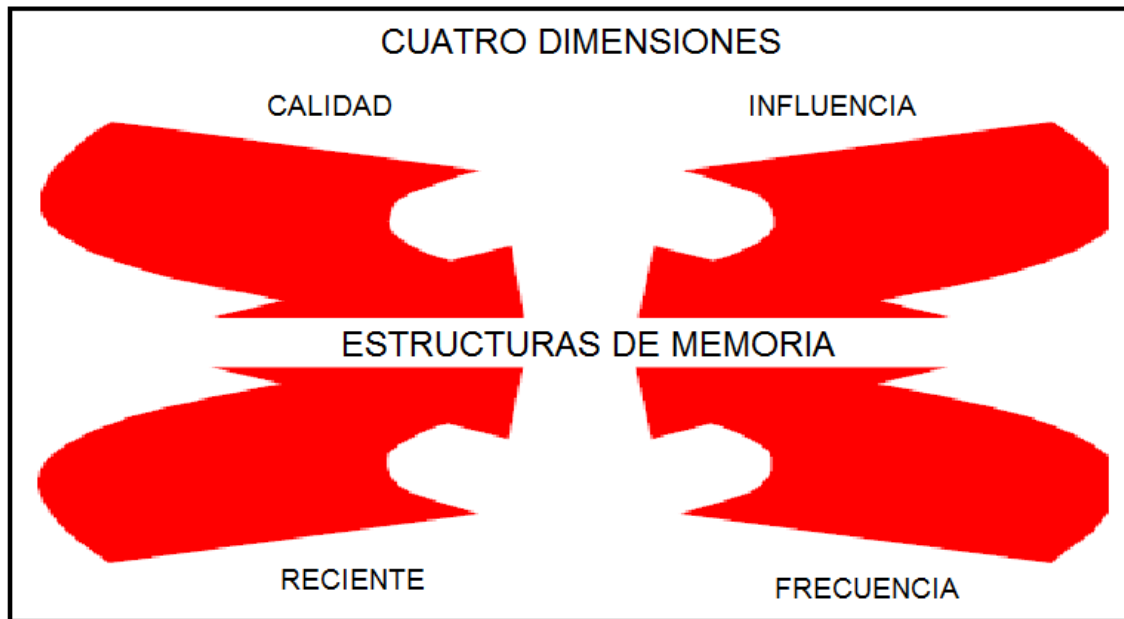


Ilustración 1 Cuatro Dimensiones de la Búsqueda Tabú

2.1.3.4.2 Intensificación y Diversificación

Las estrategias de intensificación y diversificación son componentes muy importantes dentro del Búsqueda Tabú. Para empezar, la intensificación se basa en la modificación de reglas y condiciones, de modo que se favorezca la elección de buenas combinaciones dentro de un conjunto de soluciones élite. Por otro lado, la diversificación consiste en explorar regiones no visitadas con anterioridad, lo que permite obtener soluciones diferentes a que si se hiciera una búsqueda local [GLOVER, LAGUNA, 1997].

2.1.3.4.3 Búsqueda por entorno

La búsqueda por entorno consiste en hallar un óptimo local, es aquel que es igual de bueno o mejor que las soluciones de su entorno.

En el algoritmo Búsqueda Tabú, la búsqueda por entorno tiene un significado diferente con respecto a otros algoritmos metaheurísticos ya que amplía los criterios de selección. Esto se debe a que se incluyen procedimientos constructivos y destructivos, a diferencia de los otros algoritmos. Estas estrategias son guiadas por la memoria adaptativa y permiten recorrer el universo de soluciones cuantas veces sea necesaria hasta encontrar el óptimo local [GLOVER, MELIÁN, 2003].

Método de Búsqueda en el Entorno:

Paso 1 (Inicialización)

- (A) Seleccionar una solución de arranque $x_{Actual} \leftarrow X$
- (B) Almacenar la mejor solución actual conocida haciendo $x_{Mejor} = x_{Actual}$ y definiendo $MejorCoste = c(x_{Mejor})$.

Paso 2 (Elección y finalización).

Elegir una solución $x_{Siguiete} \in N(x_{Actual})$. Si los criterios de elección empleados no pueden ser satisfechos por ningún miembro $N(x_{Actual})$, o si se aplican otros criterios de parada, entonces el método para.

Paso 3 (Actualización).

Rehacer $x_{Actual} = x_{Siguiete}$ y si $c(x_{Actual}) < MejorCoste$, ejecutar el paso 1(B). Volver al paso 2.

Tabla 2 Búsqueda por entorno

2.1.3.4.4 Estrategias basadas en memorias a corto y largo plazo

El uso de memorias a corto y largo plazo trae consigo la modificación del entorno de la solución actual. Ya que el algoritmo también mantiene un historial de estados encontrados durante el procedimiento de búsqueda, puede determinar en base experiencia las soluciones que pueden ser alcanzadas desde la solución actual.

Las estrategias basadas en memoria a corto plazo consisten en armar una estructura denominada lista tabú, la cual almacenará los movimientos realizados durante cada iteración. Esta lista indicará que estos movimientos no se podrán volver a repetir hasta dentro de cierto número de iteraciones.

A diferencia de las estrategias basadas en memoria a corto plazo, las de largo plazo consisten en armar una estructura que almacena la frecuencia con que se realizaron ciertos movimientos y mantenerlos a modo de historial [MELIÁN, GLOVER, 2004] [GALLART, ALVA, ALAMA, 2010].

2.1.3.4.5 Criterios de aspiración

Para que el algoritmo de búsqueda tabú pueda alcanzar un mejor nivel de ejecución, se emplea criterio de aspiración. Este mecanismo consiste en reemplazar ciertas restricciones tabú, de modo que se elimine la prohibición aplicada a un movimiento determinado.

Para emplear un criterio de aspiración es necesario medir la influencia e impacto que tendrá en la solución, de este modo se evalúa si realmente vale o no la pena hacer una excepción en las restricciones tabú [GLOVER, MELIÁN, 2003].



3. Estado del arte

A lo largo del tiempo, se han planteado varias formas para resolver el problema de selección y organización de proyectos. Algunos métodos de forma exacta y otros de manera aproximada, cada uno de estos con sus ventajas y desventajas. A continuación se presentan algunos métodos por los cuales se resolvió el problema:

3.1 Formas exactas de resolver el problema

En esta sección se muestra una forma de solucionar el problema de la selección de proyectos de TI.

3.1.1 *Multi-Criteria Decision Making (MCDM)*

MCDM es un estudio realizado con el fin de obtener nuevos métodos y procedimientos por medio de los cuales se puedan integrar aquellos criterios que entran en conflicto en la toma de decisiones para la selección de proyectos. De este modo se puede integrar de manera formal al proceso de planificación y gestión de proyectos de tecnologías de información.

Este estudio incluye matrices que ayudaran a la toma de decisiones en cuanto a que proyectos son más rentables que otros, de modo que según los elegido se favorezca los objetivos de la organización [International Society on Multiple Criteria Decision, 2013].

3.1.2 *Information Techonology Strategic Research (ITSR) Project*

Este proyecto proporciona una herramienta que permitirá identificar las tecnologías necesarias que puede ser empleadas a futuro en misiones de la NASA, de este modo poder afrontar de una mejor manera los desafíos que se puedan presentar al elegir una u otra tecnología.

Este proyecto trata específicamente de tener una lista de tecnologías a usar que pueden estar o no relacionadas, de modo que se puedan evaluar para su futuro desarrollo y uso. Esto se logra revisando los requisitos de una generación de proyectos y misiones que se irán a realizar, identificando las tecnologías que se pueden necesitar para cada proyecto o misión [NASA, 2013].

3.1.3 Algoritmo genético para resolver el problema de selección de proyectos

Considerando la importancia de la adecuada selección de proyectos de Tecnologías de Información, se propone un algoritmo genético que permita optimizar la ganancia, la factibilidad de ejecución, los costos y los riesgos involucrados en cada uno de los proyectos seleccionados.

Algunas de las variables a usar en este algoritmo pueden ser:

- Número de proyectos.
- Lista de proyectos.
- Costo de cada proyecto.
- Dependencia de proyectos.
- Posibles riesgos de cada proyecto.
- Beneficio estimado de cada proyecto.

Las variables mencionadas podrán ser operadas basándose en los procedimientos que involucra el algoritmo propuesto, el cual consiste en aplicar algunos operadores (operadores genéticos) de modo que se vayan generando mejores soluciones [CUEVA, TUPIA, GUANIRA, 2013].

A continuación un esquema general de un algoritmo genético [TUPIA, 2009]:

<p>Inicio Algoritmo Genético</p> <ol style="list-style-type: none"> 1. Generar Población Inicial. 2. P_Anterior = Población Inicial. 3. Mientras no se cumpla condición de parada Hacer <ol style="list-style-type: none"> 3.1 Aplicar Operador Casamiento (tasa, P_Anterior, P_Nueva). 3.2 Aplicar Operador Mutación (tasa, P_Anterior, P_Nueva). 3.3 Aplicar Operador Inversión (tasa, P_Anterior, P_Nueva). 3.4 Eliminar Aberraciones (P_Anterior, P_Nueva). 3.5 Aplicar Mecanismo Selección Aleatoria (P_Nueva, fitness). 3.6 Medir Convergencia (P_Nueva, mejor_individuo). 3.7 P_Anterior = P_Nueva. 3.8 P_Nueva = φ. <p>Fin Mientras 3. Fin Algoritmo Genético.</p>

Tabla 3 Algoritmo Genético para la selección de proyectos

3.2 Formas aproximadas de resolver el problema

En esta sección se muestran algunas formas de resolver el problema de selección y organización de proyectos en general, no solo de Tecnologías de Información.

3.2.1 *The Art of Project Selection*

Ya que la selección de proyectos se considera un proceso crítico, la correcta elección de los mismos puede mejorar de manera significativa la organización y cumplir mejor los objetivos de la misma. Para lograrlo se proponen 4 pasos a seguir:

- a) Selección de criterios de análisis: Consiste en desarrollar factores que puedan ser usados en la clasificación del proyecto y la toma de decisiones. Estos criterios pueden agruparse por objetivos, por requerimientos tecnológicos, mitigación de riesgos, etc.

- b) Definición del proyecto: Consiste en conocer el proyecto de modo que en base a sus requerimientos y problemática se puedan plantear soluciones. Entre algunos elementos a conocer para este paso se tienen:
- Nombre del proyecto.
 - Solicitante del proyecto.
 - Patrocinador del proyecto.
 - Planteamiento del problema.
 - Objetivo y expectativas.
 - Evaluación de riesgos
 - Costo estimado.
- c) Evaluación del proyecto: Esta etapa tiene como objetivo utilizar los criterios seleccionados en el primer paso de modo que los proyectos puedan ser clasificados y comparados con otros proyectos candidatos a estar en un portafolio, esto en base a puntuaciones obtenidas.
- d) Decisión del proyecto: Esta última etapa del proceso de selección es la más sencilla, ya que si se siguieron los pasos anteriores correctamente, se tienen los proyectos candidatos clasificados y comparados [SOREN, 2004].

3.2.2 Algoritmo GRASP para resolver el problema de selección de proyectos

Se sabe que el algoritmo GRASP funciona en base a una lista de candidatos que viene dada por el mejor elemento, el peor elemento y una constante alfa que permitirá relajar la solución, evitando así que la solución sea muy voraz o muy aleatoria. Para el caso de la selección de proyectos, de entre los cuales se quiere seleccionar los que mayor beneficio generen, se necesita una función objetivo la cual puede ser de la forma:

$$\text{Función (i)} = \text{Ganancia (i)} / \text{Costo (i)}$$

Esta ecuación evaluará cada proyecto y se obtendrá una proporción, lo que permitirá conocer el mejor y el peor candidato.

β = proyecto no seleccionado con mayor valor Ganancia (i) / Costo (i)

ξ = proyecto no seleccionado con menor valor Ganancia (i) / Costo (i)

Con estos valores es posible hallar la lista de candidatos (RCL):

$$RCL = \{P_i \in \text{Proyectos_No_Seleccionados} / \beta \leq \text{Ganancia (i) / Costo (i)} \leq \beta + \alpha * (\xi - \beta)\}$$

Donde para hallar el valor de α se deben utilizar tablas y experimentación numérica.

Una vez obtenida la lista de candidatos, se elige un proyecto de manera aleatoria y será seleccionado siempre y cuando no exceda el presupuesto que se tiene [TUPIA, 2009].

3.2.3 A Multiple Criteria 0-1 Linear Programming Method of Project Planning and Investment Decision Making

La complejidad para la evaluación y selección de proyectos radica en los múltiples criterios que se pueden emplear, así como la escasez de recursos. Por este motivo se recomienda usar este método de planificación en el cual intervienen variables como:

- Criterios a emplear representados por un vector, ya que pueden ser muchos criterios.
- Proyectos a evaluar representados por un vector, ya que pueden ser muchos proyectos.
- Periodo de planificación de cada proyecto, también representado por un vector.
- Tiempo de desarrollo de cada proyecto.

- Inversión total por cada proyecto.
- Tiempo total de inversión planificada.
- Ganancia otorgada por cada proyecto.
- Ganancia estimada por cada proyecto en un tiempo t .
- Recursos disponibles.

Este modelo matemático fue aplicado con éxito en la selección y planificación de proyectos de una gran empresa. Donde había más de cien proyectos entre los candidatos a seleccionar en un período de 10 años [WANG, HANG, 1997].

3.2.4 An Immune Algorithm for Optional Selection Problem of Investment Projects

Este algoritmo propuesto está basado en la simulación estocástica para dar solución general al problema de la selección de proyectos.

Al momento de calcular, este algoritmo se refiere a la solución óptima como antígeno, y a la solución factible la llama anticuerpo. El algoritmo propuesto trabaja mediante iteraciones, en las cuales se identificará el antígeno y el anticuerpo. Este algoritmo se detendrá hasta obtener el mejor anticuerpo posible [BIN, 2009].

3.3 Productos comerciales para resolver el problema

Para resolver el problema de selección y organización de proyectos existen algunos softwares, los que se listarán a continuación:

3.3.1 Augeo Software

Las soluciones desarrolladas por Augeo Software, proveen las herramientas operacionales, estratégicas y analíticas que permitirán:

- Completar los proyectos de manera exitosa.
- Elegir y seleccionar los proyectos adecuados.
- Asegurar la calidad de los proyectos.

La principal razón de éxito de este software radica en que se considera muy importante la selección de proyectos, por lo que formalizan este procedimiento de modo que los proyectos elegidos otorguen ganancia, pero sobre todo que estos proyectos se adapten a los objetivos de la organización [AUGEO, 2012].

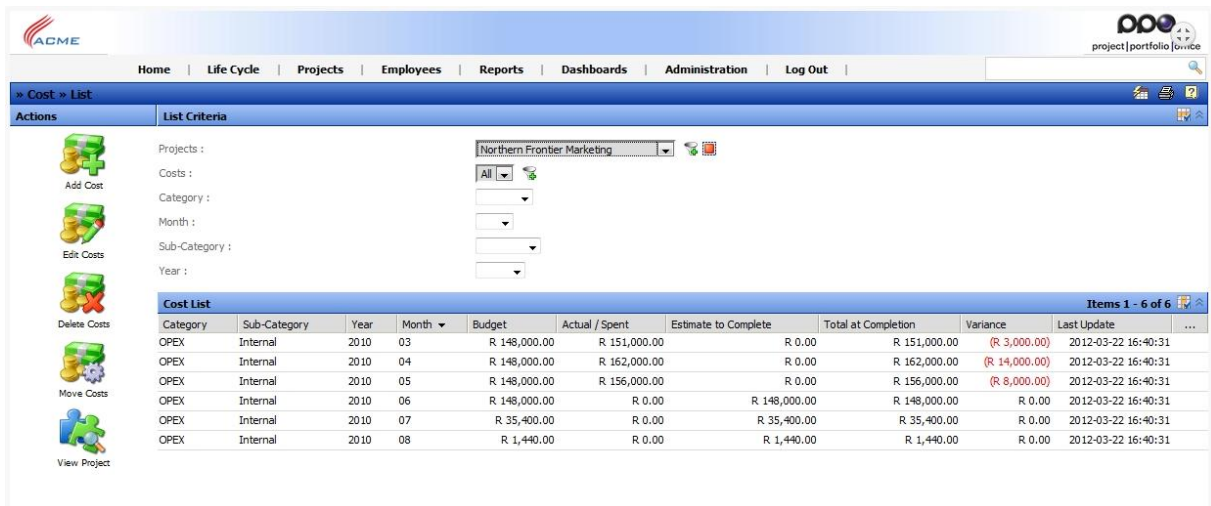
3.3.2 Project Management Office

Es una solución incluida en Project Portfolio Office (PPO) en la que intervinieron expertos en dirección de proyectos. El motivo de este software es asistir y apoyar a los equipos de trabajo, en cualquier tipo de organización, en temas de planificación, gestión, y ejecución de proyectos. Cabe mencionar que Project Portfolio Office ha ganado premios de gestión de carteras y portafolios.

La solución Project Management Office está dirigida a organizaciones que desarrollan proyectos para una o más divisiones. Esta solución permite realizar seguimientos a los proyectos. Entre sus funcionalidades se tienen [PROJECT PORTFOLIO OFFICE, 2013]:

- Gestión y priorización de portafolios o carteras.
- Gestión costo y beneficio.

- Planificación y gestión de tareas.
- Gestión de riesgos.
- Emisión de reportes cada proyecto y portafolios.



Category	Sub-Category	Year	Month	Budget	Actual / Spent	Estimate to Complete	Total at Completion	Variance	Last Update
OPEX	Internal	2010	03	R 148,000.00	R 151,000.00	R 0.00	R 151,000.00	(R 3,000.00)	2012-03-22 16:40:31
OPEX	Internal	2010	04	R 148,000.00	R 162,000.00	R 0.00	R 162,000.00	(R 14,000.00)	2012-03-22 16:40:31
OPEX	Internal	2010	05	R 148,000.00	R 156,000.00	R 0.00	R 156,000.00	(R 8,000.00)	2012-03-22 16:40:31
OPEX	Internal	2010	06	R 148,000.00	R 0.00	R 148,000.00	R 148,000.00	R 0.00	2012-03-22 16:40:31
OPEX	Internal	2010	07	R 35,400.00	R 0.00	R 35,400.00	R 35,400.00	R 0.00	2012-03-22 16:40:31
OPEX	Internal	2010	08	R 1,440.00	R 0.00	R 1,440.00	R 1,440.00	R 0.00	2012-03-22 16:40:31

Ilustración 2 Reporte de Project Management Office

3.3.3 Project Portfolio Optimization

Project Portfolio Optimization utiliza herramientas y métodos para cuantificar el costo y beneficio de los proyectos que se están ejecutando o se irán a ejecutar. Del mismo modo, mide el riesgo de los proyectos para cuantificar rentabilidad/riesgo.

Para lograrlo se ingresa en un formulario las prioridades de la organización, así como los recursos con los que se cuenta. De este modo se podrán calcular las limitaciones y priorizar los proyectos de modo que cumplan los objetivos de la organización. Ya con los datos necesarios dentro del software, se tendrá acceso a reportes que permitirán tomar una decisión, de modo que se maximice la ganancia de la empresa [PWC, 2012-2013].

Label	Unit	Portfolio summary
Number of projects		241
Cost (CY)	\$M	321.4
Total cost (PV)	\$M	817.5
Capital projects		194
Maintenance projects		47
Total cost of capital projects (PV)	\$M	752.3
Total cost of O&M projects (PV)	\$M	65.2
Mandated projects		14
Total cost of mandated projects (PV)	\$M	41.6
NPV (incl. all benefits)	\$M	932.0

Ilustración 3 Reporte de Project Portfolio Optimization

3.4 Productos de investigación para resolver el problema

En esta sección se mostrarán algunos aportes de investigadores que tienen como objetivo solucionar el problema de selección y organización de proyectos:

3.4.1 A Research on Multiple Projects Selection Base on Matching Matrix Model

Siendo el principal problema, elegir y seleccionar el proyecto adecuado de entre una gran cantidad de los mismos, los autores de esta investigación proponen un modelo que se basa en tomar los factores críticos de éxito de la inversión de los proyectos en consideración, integrando las condiciones de restricción de la organización, para luego aplicar el método de Coincidencia de Matriz (MatchingMatrix, propuesto por Gold y Campbell) para así evaluar y seleccionar los proyectos en base al *parenting opportunity*.

Parenting opportunity se refiere a la posibilidad de mejora de un negocio, es decir, si se puede mejorar más. Para esto se analizan las oportunidades que la empresa u organización tiene para darle mayor valor a su negocio [GAN, GAO, 2011].

3.4.2 The research of the financial Evaluation Methods of investment Projects Based on EVA

Actualmente, debido a la diversificación de la inversión, se esperan mayores beneficios y ganancias de los proyectos en los que se invierte. Esto ocasiona que la competencia entre proyectos sea mayor y sea más complicado elegir los adecuados para un portafolio.

Para el problema planteado, se propone un nuevo método de evaluación basado en EVA. Para empezar, se debe que el EVA es una medida que real para el valor económico de la empresa y que puede expresarse como, la ganancia real después de haberle restado el costo de capital invertido. Dicho esto, el método propuesto consiste en obtener el flujo de caja cada vez que se deduce el costo total de capital basándose en EVA, esto siempre teniendo en cuenta la viabilidad del proyecto.

Por tanto, si se tuvieran muchos proyectos para armar un portafolio, este método proporcionará un punto de vista más adecuado a los inversores, de modo que puedan elegir los mejores proyectos para su portafolio [LI-HUA, WEN-HUA, 2011].

3.5 Conclusiones sobre el estado del arte

Ya presentadas algunas formas de resolver el problema de selección y organización de proyectos se observa que es muy complicado realizarlo, ya que los factores a analizar pueden ser muchos, lo que añade complejidad computacional al problema. Las soluciones aproximadas, si bien otorgan una visión general del problema a tratar, no se enfocan específicamente en la selección de proyectos de TI y como se ha visto pueden ser de una suma complejidad matemática. Por otro lado, los productos comerciales son una buena opción; sin embargo, son software que no están hechos a la medida de la organización por lo que pueden inservibles en algunos casos. Por tanto, el presente proyecto de fin de carrera, aportará un método sencillo que permita realizar y seleccionar los proyectos adecuados de TI que permitirán crecer a la organización. Ya que es una adaptación algorítmica puede ser aplicado en cualquier organización.

CAPÍTULO 2

1. Objetivo general

Implementar un algoritmo de Búsqueda Tabú para resolver el problema de la selección de proyectos de TI.

2. Objetivos específicos

- Objetivo 1: Desarrollar una herramienta que permita la carga masiva de datos a procesar por los algoritmos GRASP construcción y Búsqueda Tabú.
- Objetivo 2: Definir la arquitectura de información para el algoritmo de Búsqueda Tabú, el cual incluirá las estructuras de datos e información que permitirán la posterior implementación del algoritmo.
- Objetivo 3: Definir arquitectura de información para el algoritmo de GRASP, el cual incluirá las estructuras de datos e información que permitirán la posterior implementación del algoritmo.
- Objetivo 4: Definir la función objetivo a ser usada en los algoritmos GRASP y Búsqueda Tabú, considerando variables propias del problema como son costo, beneficio y criticidad de riesgos. En ambos casos, la función objetivo será la misma.
- Objetivo 5: Implementar un algoritmo GRASP el cual servirá, en primera instancia para la comparación y posteriormente para la generación de la población inicial a ser utilizada por el algoritmo de Búsqueda Tabú.
- Objetivo 6: Seleccionar y desarrollar la experimentación numérica adecuada para el problema para así comparar los resultados de los algoritmos GRASP y Búsqueda Tabú mediante herramientas estadísticas.
- Objetivo 7: Desarrollar una interfaz que muestre los resultados obtenidos por los algoritmos GRASP y Búsqueda Tabú.

3. Resultados esperados

- Resultado 1 para el objetivo 1: Herramienta generadora de datos que arrojará un archivo que servirá como input al algoritmo GRASP.
- Resultado 2 para el objetivo 2: Estructura de datos que soporte la implementación del algoritmo de Búsqueda Tabú.
- Resultado 3 para el objetivo 3: Estructura de datos que soporte la implementación del algoritmo de GRASP.
- Resultado 4 para el objetivo 4: Función objetivo a ser utilizada en los algoritmos GRASP y Búsqueda Tabú tomando como base las consideraciones propias del problema de la selección de proyectos.
- Resultado 5 para el objetivo 5: Algoritmo GRASP construcción implementado.
- Resultado 6 para el objetivo 6: Diseño de la experimentación numérica e interpretación de resultados de la misma.
- Resultado 7 para el objetivo 7: Interfase que muestra la comparación de los resultados de los algoritmos GRASP y Búsqueda Tabú.

4. Herramientas, métodos y procedimientos

A continuación se presentan las herramientas, métodos y procedimientos a usar en el presente proyecto de fin de carrera.

4.1 Mapeo



	Extreme Programming	Netbeans	Metodología GRASP	Metodología Búsqueda Tabú	Comparación de dos tratamientos	Java	Microsoft Excel
RE1: Herramienta generadora de datos	x						
RE2: Estructura de datos Búsqueda Tabú	x	x					
RE3: Estructura de datos GRASP construcción	x	x					
RE4: Función objetivo a ser utilizada en los algoritmos GRASP y Búsqueda Tabú			x	x			
RE5: Algoritmo GRASP construcción implementado	x	x	x				
RE6: Diseño de la experimentación numérica e interpretación de resultados de la misma.					x		x
RE7: Interfase que muestra la comparación de los resultados de los algoritmos GRASP y Búsqueda Tabú	x	x				x	

Tabla 4 Mapeo de Herramientas

4.2 Herramientas para el desarrollo

a) Netbeans

Netbeans es un IDE (Integrated Development Environment) desarrollado para escribir, compilar y ejecutar programas. Para esto cuenta con una interfaz que permite editar código de una manera rápida, así como la sencilla administración de proyectos de programación que se tienen [NETBEANS, 2013].

Debido a que Netbeans es una herramienta que permite administrar los proyectos de programación de una manera sencilla se justifica el uso de esta herramienta.

Otra de las razones es que cuenta con funcionalidades que permiten la fácil edición del código, como autocompletar, lo que ayudará significativamente en la implementación de los algoritmos. Finalmente, el uso de esta herramienta también está relacionado con el lenguaje de programación Java a usar en el proyecto, ya que este IDE fue desarrollado principalmente para hacer uso de este lenguaje.

b) Java

Java es un lenguaje de programación orientado de objetos mediante el cual se pueden escribir programas. Este lenguaje es simple, distribuido, interpretado, sólido, seguro, portable y multiplataforma (funciona en varios sistemas operativos) [JAVA, 2013].

Debido a que Java es un lenguaje de programación multiplataforma y es adecuado para el IDE Netbeans, ya que fue desarrollado para este principalmente, se justifica la elección. Otra razón que justifica el uso de este lenguaje de programación es el conocimiento del mismo.

4.3 Herramientas para la gestión

Para la adecuada gestión del proyecto se usará la metodología desarrollada por el Project Management Institute (PMI), Project Management Body of Knowledge (PMBOK) o Guía de Fundamentos para la Dirección de proyectos.

PMBOK posee 5 Grupos de Procesos, los cuales son:

- Grupo del Proceso de Iniciación.
- Grupo del Proceso de Planificación.
- Grupo del Proceso de Ejecución.
- Grupo del Proceso de Seguimiento y Control.
- Grupo del Proceso de Cierre.

Asimismo, se tienen 9 Áreas de Conocimiento, las cuales son:

- Gestión de la Integración del Proyecto.
- Gestión del Alcance del Proyecto.
- Gestión del Tiempo del Proyecto.
- Gestión de los Costos del Proyecto.
- Gestión de la Calidad del Proyecto.
- Gestión de los Recursos Humanos del Proyecto.
- Gestión de las Comunicaciones del Proyecto.
- Gestión de los Riesgos del Proyecto.
- Gestión de las Adquisiciones del Proyecto

Tanto las los Grupos de Procesos como las Áreas de Conocimiento se relacionan entre sí a lo largo de todo el proyecto.

Al momento de emplear la metodología PMBOK se debe tener en cuenta que debido a la naturaleza del presente proyecto de fin de carrera no se usará toda la metodología sino solo aquellas partes que apliquen y se adapten mejor a las necesidades del mismo. [PMBOK, 2009]

La razón por la que se elige esta metodología es la flexibilidad y adaptabilidad que tiene para los diferentes tipos de proyectos. Otro motivo es que está orientado al seguimiento del proyecto y uso de herramientas de manera que se pueda conseguir el objetivo del mismo.



GRUPO DE PROCESO	ÁREA DE CONOCIMIEN	PROCESO	OBJETIVO DEL PROCESO	JUSTIFICACIÓN
Grupo del Proceso de Iniciación	Gestión de la Integración del Proyecto	Desarrollar del acta de constitución del proyecto	Desarrollar el documento de autorización del proyecto. Contiene requerimientos iniciales	Tanto el asesor como el profesor del curso deben estar de conformes con el proyecto para su iniciación
		Desarrollar el Plan para la Dirección del Proyecto	Documentar acciones necesarias para preparar, integrar y coordinar los planes	Todos los documentos presentados en el proyecto deben ser coherentes y de ayuda para conseguir el objetivo del mismo
Grupo del Proceso de Planificación	Gestión del Alcance del Proyecto	Definir el alcance	Desarrollar descripción detallada del proyecto	Se debe tener claro el alcance del proyecto con el fin de que los esfuerzos sean productivos para conseguir el objetivo. De igual modo sucede con la división de entregables
		Crear la EDT (Estructura de Desglose de Trabajo)	Subdividir el trabajo en componentes más fáciles de dirigir	
	Gestión del Tiempo del Proyecto	Definir las actividades	Identificar las acciones específicas a ser realizadas para elaborar los entregables del proyecto	Se requiere organizar el tiempo de modo que se puedan cumplir los objetivos en el plazo establecido haciendo un adecuado uso de los recursos que se tienen
		Estimar la duración de las actividades	Establecer aproximadamente el tiempo de trabajo necesario para finalizar cada actividad	
		Desarrollar el Cronograma	Analizar el orden de actividades, su duración y restricciones para crear el cronograma del proyecto	

GRUPO DE PROCESO	ÁREA DE CONOCIMIEN	PROCESO	OBJETIVO DEL PROCESO	JUSTIFICACIÓN
	Gestión de las Comunicaciones del Proyecto	Planificar las Comunicaciones	Determinar las necesidades de información de los interesados en el proyecto y para definir cómo abordar las comunicaciones	Se requiere comunicación con los profesores que puedan aportar al proyecto por ser especialistas en la rama del proyecto
		Planificar la Gestión de Riesgos	Definir cómo realizar las actividades de gestión de riesgos para un proyecto	
	Gestión de los Riesgos del Proyecto	Identificar Riesgos	Determinar los riesgos que pueden afectar el proyecto	Se requiere una adecuada gestión de riesgos de modo que aquellos riesgos negativos puedan ser controlados y los positivos explotados
		Realizar Análisis Cuantitativo de Riesgos	Analizar el efecto de los riesgos identificados sobre los objetivos generales del proyecto	
		Planificar la Respuesta a los Riesgos	Desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto	
Grupo de Proceso de Ejecución	Gestión de la Integración de Proyecto	Dirigir y Gestionar la Ejecución del Proyecto	Ejecutar el trabajo definido en el plan para la dirección del proyecto para cumplir con los objetivos del proyecto	Se requiere seguir un orden para llevar a cabo el proyecto de la mejor manera

GRUPO DE PROCESO	ÁREA DE CONOCIMIENTOS	PROCESO	OBJETIVO DEL PROCESO	JUSTIFICACIÓN
Grupo del Proceso de Seguimiento y Control	Gestión de la Integración del Proyecto	Dar Seguimiento y Controlar el Trabajo del Proyecto	Revisar, analizar y regular el avance a fin de cumplir con los objetivos de desempeño definidos en el plan para la dirección del proyecto	Se requiere supervisión por parte del asesor con el fin de verificar que se estén cumpliendo los objetivos del proyecto.
	Gestión del Alcance del Proyecto	Verificar el Alcance	Formalizar la aceptación de los entregables del proyecto que se han completado	Se requiere que el asesor apruebe cada entregable y verifique que los objetivos no excedan el alcance del proyecto. En caso suceda esto, se debe controlar y direccionar los esfuerzos a cumplir con el alcance establecido
		Controlar el Alcance	Dar seguimiento al estado del alcance del proyecto y del producto, y se gestionan cambios a la línea base del alcance	
	Gestión del Control de Tiempo del Proyecto	Controlar el Cronograma	Dar seguimiento a la situación del proyecto para actualizar el avance del mismo y gestionar cambios a la línea base del cronograma	Se requiere un control del cronograma del proyecto para saber si hay retrasos con respecto al tiempo establecido
	Gestión de las Comunicaciones del Proyecto	Informar el Desempeño.	Recopilar y distribuir la información sobre el desempeño, incluidos informes de estado, mediciones del avance y proyecciones	Se requiere que el asesor supervise el desempeño del tesista con el fin de que el proyecto se lleve a cabo en los tiempos establecidos
	Gestión de los Riesgos del Proyecto	Dar Seguimiento y Controlar los Riesgos.	Dar seguimiento a los riesgos residuales, identificar nuevos riesgos y evaluar la efectividad de los controles implementados	Se requiere tener mapeados los nuevos riesgos que puedan surgir durante la ejecución del proyecto
Grupo del Proceso de Cierre	Gestión de la Integración del Proyecto	Cerrar el Proyecto o Fase	Finalizar todas las actividades a través de todos los grupos de procesos de dirección de proyectos para completar formalmente el proyecto o una fase del mismo	Se requiere que tanto el asesor como el profesor del curso aprueben el producto final obtenido

Tabla 5 Herramientas para la gestión

4.4 Herramientas Metodológicas

Para conseguir cumplir con los objetivos del proyecto es necesario emplear algunas metodologías, entre las cuales se tienen:

a) Extreme Programming (XP)

Es una metodología de desarrollo desarrollada por Kent Beck. La particularidad de esta metodología es que se centra más en programación que en la documentación.

XP tiene como principales valores a la simplicidad, comunicación, retroalimentación (feedback) y coraje [BECK, ANDRES, 2004].

Asimismo consta de varias etapas como [TUPIA, 2005]:

- Prototipos basados en una lista de tareas.
- Simplicidad de diseño.
- Validación y pruebas.
- Integración constante.

La razón de usar esta metodología es que al estar ante un proyecto de tesis de desarrollo de algoritmos, es preferible enfocarse más en la programación que en la documentación. Esta metodología será usada tanto en la implementación de una herramienta de carga masiva de datos como en la implementación de los algoritmos GRASP construcción y Búsqueda Tabú.

b) Comparación de dos tratamientos

Es un test estadístico usado comúnmente para la comparación de dos o más medias de alguna población.

- **Prueba de Kolmogorov-Smirnov**

“La prueba de Kolmogorov-Smirnov para una muestra es un procedimiento de "bondad de ajuste", que permite medir el grado de concordancia existente entre la distribución de un conjunto de datos y una distribución teórica específica. Su objetivo es señalar si los datos provienen de una población que tiene la distribución teórica especificada, es

decir, contrasta si las observaciones podrían razonablemente proceder de la distribución especificada [UNIVERSITAT DE VALENCIA, 2010]. ”

La razón de usar esta prueba es asegurar que los resultados obtenidos se comportan de manera normal y así poder aplicar la prueba estadística Z para la comparación de dos muestras.

- **Prueba F de Fisher**

“Se emplea para probar si dos muestras provienen de poblaciones que poseen varianzas iguales. Esta prueba es útil para determinar si una población normal tiene mayor variación que la otra y también se aplica cuando se trata de comparar simultáneamente varias medias poblacionales. La comparación simultánea de varias medias poblacionales se conoce como análisis de varianza”. [RETURETA, 2010]

La prueba F de Fisher será usada para garantizar la calidad de las soluciones ya que analiza la variación de las poblaciones, se justifica la elección de esta.

- **Prueba Z**

“Se basa en la diferencia entre las proporciones de dos muestras. Bajo la hipótesis nula, se supone que las proporciones de dos poblaciones son iguales y ya que el conjunto estimado para las proporciones se basa en esta hipótesis, se combinan las proporciones de las dos muestras para calcular una estimación global de la proporción de la población común. Esta prueba puede ser usada para la diferencia entre las proporciones de dos muestras para determinar si existe una diferencia en la proporción de éxitos de los dos grupos (prueba de dos colas) o si un grupo tiene mayor proporción de éxitos que el otro grupo (prueba de una cola) [BERENSON, LEVINE, KREHBIEL, 2006].”

La justificación para emplear esta prueba radica en que para comparar los algoritmos, tanto el GRASP como el de Búsqueda Tabú, es necesario usar un número determinado de muestras. Ya que serán 40 muestras las que se evaluarán, es imprescindible esta prueba. De haber sido menos de 35 muestras la prueba a elegir sería la de T-Student.

c) Metodología del algoritmo GRASP

“La Metodología GRASP para construir una solución en lugar de considerar apenas un candidato disponible para el seleccionado (criterio usado por el algoritmo goloso), crea un conjunto de candidatos del que seleccionará uno de ellos aleatoriamente. Por tal motivo, la metodología permite construir varias soluciones diferentes” [GANOZA, SOLANO, 2004].

Cabe mencionar que dentro de esta metodología está incluida la calibración del alfa, que es la constante de relajación del algoritmo GRASP construcción a implementar para la comparación con el algoritmo de Búsqueda Tabú.

Se justifica el uso de la metodología ya que es necesario implementar este algoritmo para la posterior comparación con el algoritmo de Búsqueda Tabú.

d) Metodología del algoritmo Búsqueda Tabú

La metodología del algoritmo de Búsqueda Tabú viene dado por: Dado una solución perteneciente a un universo de soluciones, se evalúan y se mueve a la mejor solución vecina, sin embargo se considera el vecindario reducido dentro de los cuales solo algunas soluciones están disponibles (no tabú).

Ya que se el presente proyecto de fin de carrera consiste en adaptar el algoritmo de Búsqueda Tabú a la selección de proyectos, es necesario usar la metodología correspondiente a la implementación de dicho algoritmo.

4.5 Otras herramientas

- **Microsoft Excel**

Microsoft Excel es una aplicación perteneciente a Microsoft Office 2010. Es un programa que consiste en hojas de cálculo en las cuales se pueden realizar trabajos de cálculo tales como préstamos, informes de ventas y estadística. Del mismo modo, permite la programación de macros en VBA (Visual Basic for Application). Estas funcionalidades permiten ahorrar tiempo, simplificar el trabajo y aumentar la productividad [MICROSOFT OFFICE, 2010].

Se justifica el uso de esta herramienta ya que, gracias a que permite la elaboración de macros, será posible implementar un programa que sirva como herramienta para la carga masiva de datos que será el input para el algoritmo GRASP. Asimismo permitirá desarrollar la interpretación de la experimentación numérica ya que cuenta con funciones estadísticas para realizar esta evaluación.

5. Alcance

Este proyecto pretende apoyar en la toma de decisiones de selección de proyectos de TI, es decir, seleccionar un portafolio de proyectos que permita a la organización obtener ganancias.

Para lograr este objetivo se tomarán en cuenta ciertas variables, las cuales pueden ser consideradas las más importantes para la adecuada selección de proyectos. Estas variables son:

- Costo del proyecto: Es el costo aproximado que tendrá la implementación del proyecto, es decir, la inversión que requiere.
- Beneficios del proyecto: Es el beneficio aproximado que se obtendrá cuando el proyecto se haya realizado completamente.
- Tiempo de duración: Es el tiempo aproximado que durará la implementación y ejecución del proyecto.
- Riesgos del proyecto: Son los riesgos que puede tener cada proyecto. Estos riesgos pueden ser críticos, altos, relevantes, moderados y bajos. Cada una de estas criticidades tendrá un valor asignado de modo simule que tan riesgoso es el proyecto.

Una vez definidas las variables que interactuarán con el proyecto, se debe establecer una función objetivo, la cual será la misma para los algoritmos GRASP y Búsqueda Tabú. Del mismo modo se deben identificar aquellas estructuras de datos que formarán parte de la arquitectura de información a ser usada en el proyecto. Luego, se debe desarrollar la aplicación para finalmente realizar la experimentación numérica, la cual permitirá la comparación entre los dos algoritmos planteados, demostrando, en este caso, que el algoritmo de Búsqueda Tabú es mejor que el GRASP.

5.1 Limitaciones

Entre las limitaciones del proyecto se tienen:

- El tiempo de ejecución puede variar dependiendo de las características del hardware y software de la computadora en que se ejecute. Esto se debe a la complejidad de los algoritmos y la cantidad de veces que deben ser ejecutados para obtener resultados.
- No se podrán agregar más variables a las ya establecidas (costo, beneficio, tiempo y riesgos).
- Los resultados del proyecto pueden verse afectados por la cantidad y calidad de datos de entrada.
- El proyecto estará aplicado únicamente a los proyectos de Tecnologías de Información.

5.2 Riesgos

En la siguiente tabla se muestran los riesgos identificados que pueden afectar el presente proyecto de fin de carrera.

RIESGO IDENTIFICADO	IMPACTO EN EL PROYECTO	MEDIDAS CORRECTIVAS PARA MITIGAR
Mala planificación del proyecto.	<ul style="list-style-type: none"> • Entregables revisados a destiempo. • Rechazo de entregables. • No realizar los entregables. 	Revisar los plazos y tiempos de modo que estos se puedan ajustar para cumplir con el proyecto.
Pérdida parcial o total de información y avances del proyecto.	<ul style="list-style-type: none"> • No presentar entregables. • Pérdida de tiempo ya que se tendrá que volver a empezar. • Desaprobar el curso. 	Realizar respaldo de información de manera periódica.
Falta de comunicación con el asesor.	<ul style="list-style-type: none"> • Entregables no pueden ser corregidos y por tanto tampoco aprobados. • Ya que no hay asesoría se pueden cometer errores graves en el proyecto. 	Comunicar oportunamente al profesor del curso para asignación de otro asesor.

Tabla 6 Riesgos del proyecto

6. Justificación y viabilidad

6.1 Justificativa del proyecto de tesis

En la actualidad, las Tecnologías de Información juegan un papel muy importante en las empresas ya que son necesarias para el crecimiento y desarrollo de las mismas. Esto se debe a que existe la necesidad de desarrollo de sistemas, integración de información, automatización de procesos, etc. Sin embargo, estos proyectos deben estar priorizados ya que no se pueden desarrollar todos, esto se debe a que los tiempos para llevarlos puede ser insuficiente y los presupuestos son por periodos anuales. Es importante tener esto en cuenta, pues es conocido que las empresas deben planificar y ajustar presupuestos para un año y seleccionar en que proyectos se invertirá. Otro punto importante es la interdependencia, ya que en muchos casos para desarrollar e implementar un proyecto es necesario tener desarrollado otro proyecto con anterioridad que haga viable y posible la implementación y desarrollo del último. Finalmente, si bien en la actualidad existen muchos métodos exactos para la adecuada selección de proyectos, estos tomarían demasiado tiempo, a diferencia de la solución propuesta en el presente proyecto, que si bien no es exacta, se obtendrán soluciones aceptables.

Por ello, es necesaria la adecuada selección de proyectos que ayuden a la empresa a alcanzar sus objetivos estratégicos de modo que puedan obtener un máximo beneficio. Por lo que el presente proyecto de fin de carrera servirá como apoyo en la toma de decisiones al momento de seleccionar un portafolio de proyectos, el cual consistirá en un grupo de proyectos seleccionados de entre muchos otros de modo que se obtenga el mayor beneficio posible considerando los presupuestos y tiempos de desarrollo de cada proyecto.

6.2 Análisis de viabilidad del proyecto de tesis

A continuación se presentarán factores que influirán en la viabilidad del proyecto:

a) Viabilidad Técnica

En cuanto a la viabilidad técnica se tienen los conocimientos necesarios para alcanzar los objetivos del presente proyecto, son los que se han venido desarrollando en los últimos semestres de la carrera de Ingeniería Informática. Adicionalmente se contará con asesoría tanto del asesor como de los profesores que tienen dominio del tema a presentar. Cabe mencionar que las herramientas a utilizar son conocidas por el tesista, en este caso dichas herramientas serían el entorno de desarrollo Netbeans (versión 7.3) y el lenguaje de programación Java (versión 1.7).

Por otro lado, se está proponiendo una serie de metodologías adecuadas para cumplir con cada uno de los resultados esperados. Estas metodologías serán una guía y apoyo para cumplir con los objetivos del proyecto.

Finalmente, las herramientas a utilizar en el presente proyecto son de fácil acceso ya que son gratuitas, por lo que no será impedimento para el desarrollo normal del proyecto.

b) Viabilidad Temporal

En el aspecto temporal si bien el alcance no es tan reducido, la implementación del proyecto se empezará inmediatamente después de culminar el primer curso de tesis 1, obteniendo así un mayor tiempo para culminar el proyecto.

c) Viabilidad económica

En este aspecto, el presente proyecto no tendrá costos ya que las herramientas y material necesario son recursos gratuitos o educativos, como son los casos del entorno de desarrollo Netbeans y el lenguaje de programación Java.

En conclusión, a pesar de que hay algunas limitaciones y riesgos para la elaboración del proyecto, así como algunos costos menores, se cuenta con las herramientas, conocimientos y metodologías adecuadas para el desarrollo del proyecto por lo que su desarrollo es perfectamente viable.



CAPÍTULO 3

1. Herramienta generadora de datos

A continuación se presenta el modo de funcionamiento de la herramienta generadora de datos, la cual generará un archivo que servirá de “input” para las pruebas de los algoritmos GRASP construcción y Búsqueda Tabú.

Se puede configurar las variables involucradas en el problema. Estas son el número de proyectos, el presupuesto, el rango de valores del costo estimado, el rango de valores del beneficio estimado y el rango de valores del tiempo estimado (Ilustración 1).

Número de Proyectos	310				
Presupuesto	110000				
	Min	Max			
Rango Costo estimado	3000	10000			
Rango Beneficio estimado	3500	12000			
Rango Tiempo estimado	1	12			

Generar data

Ilustración 4 Pantalla de herramienta generadora de datos

Una vez que se tienen los valores para cada caso, se debe pulsar el botón “Generar data”, el cual generará un archivo con nombre “input.csv” de texto plano para ser usado en los algoritmos propuestos (Ilustración 2).

En la primeras fila se muestra el número de proyectos de TI y el presupuesto con el que se cuenta, las filas siguientes contienen los datos de cada proyecto las cuales son costo estimado, beneficio estimado, tiempo estimado, riesgo estimado y dependencia. Cabe mencionar que si la dependencia es el número 0, quiere decir que el proyecto no tiene dependencia.

310	110000			
6733.97	8425.91	8.76	3	0
3098.12	9966.15	9.52	9	0
5898.23	10832.26	1.5	8	0
9100.12	11571.23	11.58	4	0
3374.53	8535.9	9.44	7	0
7534.75	10553.31	7.85	6	0
9376.75	9408.48	11.85	6	0
3744.59	11995.02	6.87	8	0
3700.37	4375.69	7.33	5	0
5070.41	6747.09	1.5	4	0
5809.62	5865.38	11.78	2	201
5889.37	9558.21	5.51	6	0
4302.09	8458.55	3.28	4	0
4829.58	10174.3	10.96	7	0
7422.2	8834.96	11.11	3	174
9396.02	10595.95	8.64	1	0
6011.83	9262.56	11.08	6	0
5474.31	6941.09	6.09	3	0
9853.55	11477.63	3.68	4	0
3751.63	10163.96	5.3	7	0
8829.11	9084.97	7.56	4	0

Tabla 7 Estructura herramienta generadora de datos

Finalmente, para la adecuada generación de datos se consideran los siguientes aspectos:

- El beneficio estimado de cada proyecto es mayor que el costo estimado de cada proyecto.
- La ejecución de un proyecto puede depender de otro proyecto, más no de sí mismo.

CAPÍTULO 4

1. Estructuras de datos

Partiendo del supuesto que se cuentan con las variables necesarias para llevar a cabo la selección de proyectos, que son:

VARIABLE	REPRESENTACIÓN
Cantidad de proyectos	N
Presupuesto para proyectos	V
Costo estimado del proyecto i	C_i
Beneficio estimado del proyecto i	B_i
Tiempo estimado del proyecto i	T_i
Riesgo estimado del proyecto i	R_i

Tabla 8 Variables del problema

Se plantean a continuación las siguientes estructuras de datos:

N: Número de proyectos $P_1 P_2 \dots P_N$

Arreglo RCL: Arreglo bidimensional, ordenado descendientemente, donde cada entrada contiene número del proyecto y la función objetivo obtenida con las variables del mismo (costo estimado del proyecto, beneficio estimado del proyecto, tiempo estimado del proyecto y riesgo estimado del proyecto). Este arreglo se obtiene luego de obtener los rangos inferior y superior usando la constante de relajación alfa.

P	8	15	7	3	16	25	10
FO	7.4	7.1	6.3	6.2	6	6.1	5

Tabla 9 Estructuras de Datos (RCL)

Matriz P: [$P_{N \times 5}$]: Matriz de N proyectos donde cada entrada contiene número del proyecto, costo estimado del proyecto, beneficio estimado del proyecto, tiempo estimado del proyecto y riesgo estimado del proyecto)

P	1	2	3	4	5	...	N
C	2000	5000	1000	8000	15000	...	20000
B	5000	8000	2500	15000	25000	...	50000
T	10	20	5	18	14	...	16
R	3	2	1	6	4	...	8

Tabla 10 Estructuras de Datos (Matriz de proyectos)

A continuación se detalla la criticidad de los riesgos con su correspondiente valor, los mismos que irán en la matriz anterior en R.

CRITICIDAD	RANGO DE VALORES
Bajo	1 – 2
Moderado	3 – 4
Relevante	5 – 6
Alto	7 – 8
Crítico	9 – 10

Tabla 11: Tabla de estados de riesgo (R)

Matriz D: [$D_{N \times 2}$]: Matriz de N proyectos donde cada entrada contiene el número de proyecto y su correspondiente dependencia, la cual es otro número de proyecto en caso lo hubiera. En caso su correspondiente valor sea 0, quiere decir que dicho proyecto no tiene dependencia alguna.

P	1	2	3	4	5	...	N
D	3	4	1	0	12	...	8

Tabla 12 Estructuras de datos (Matriz de dependencia de proyectos)

Matriz R: $[R_{N \times N}]$: Matriz o lista tabú de $N \times N$ basada en lo reciente (corto plazo) donde cada entrada contiene el número de veces que un movimiento o intercambio está prohibido o es tabú. Este número irá disminuyendo en cada iteración hasta llegar a 0, lo que significa que ya se puede usar ese movimiento o intercambio nuevamente. Siendo un movimiento o intercambio el pasar un número de proyecto del conjunto solución a la lista P de proyectos disponibles y pasar un proyecto de la lista P al conjunto solución.

	1	2	3	4	5	6	...	N
1		0	8	0	0	2	0	0
2			0	4	0	0	0	3
3				0	0	0	5	0
4					0	0	0	7
5						1	0	0
6							0	2
...								0
N								

Tabla 13 Estructuras de datos (Memoria de lo reciente)

Matriz F: $[F_{N \times N}]$: Matriz o lista tabú de $N \times N$ basada en lo frecuente (largo plazo) donde cada entrada contiene el número de veces que un movimiento o intercambio se ha llevado a cabo. Cuando un movimiento se ha llevado a cabo muchas veces puede indicar que la solución está en un óptimo local, por lo que se le pondrá una penalización de X turnos en la lista tabú basada en lo reciente con la finalidad de explorar otras soluciones.

Siendo un movimiento o intercambio el pasar un número de proyecto del conjunto solución a la lista P de proyectos disponibles y pasar un proyecto de la lista P al conjunto solución.

	1	2	3	4	5	6	...	N
1		1	8	4	0	1	0	5
2			0	7	2	0	0	3
3				0	0	10	7	0
4					6	0	3	9
5						1	0	0
6							0	15
...								4
N								

Tabla 14 Estructuras de datos (Memoria de lo frecuente)



CAPÍTULO 5

1. Función objetivo

Ya que los algoritmos pretenden producir el mejor resultado posible dentro un conjunto de candidatos, se requiere una función objetivo. Esta función objetivo establecerá el mejor criterio de selección, la cual viene dada por:

$$\text{Max} \sum_{i=1}^n \frac{B_i}{C_i * T_i * R_i}$$

Donde:

B_i : Beneficio estimado del proyecto i

C_i : Costo estimado del proyecto i.

T_i : Tiempo estimado del proyecto i (en meses).

R_i : Riesgo estimado del proyecto i.

Como se puede observar el objetivo de esta función es que sea maximizada. Esto a razón de que se quiere:

- Obtener mayor beneficio B_i
- Reducir costo C_i
- Reducir tiempos T_i
- Reducir riesgos R_i

Por ello el criterio de selección será elegir aquel proyecto que tenga mayor valor y así obtener un portafolio que permita a la organización generar mayor retorno de ganancia.

Finalmente, se debe tener en cuenta las siguientes restricciones:

- Presupuesto: ya que se cuenta con un presupuesto (V) para cubrir los proyectos, este no podrá ser sobrepasado por el costo acumulado del portafolio.
- Dependencia: representada por la Matriz D, la cual contiene los proyectos requeridos para realizar la ejecución de algún otro proyecto.



CAPÍTULO 6

1. Pseudocódigo de algoritmo GRASP construcción

Tener en consideración las siguientes variables:

VARIABLE	REPRESENTACIÓN
Cantidad de proyectos	N
Presupuesto para proyectos	V
Riesgo estimado del proyecto i	R_i

Tabla 15 Variables para algoritmo GRASP construcción

Adicionalmente, considerar los arreglos:

ARREGLO	REPRESENTACIÓN
Matriz de Proyectos	P
Matriz de Dependencia	D
Arreglo RCL	RCL
Arreglo Solución	S

Tabla 16 Estructuras de datos para algoritmo GRASP construcción

A continuación se presenta el pseudocódigo del algoritmo de GRASP construcción:

Inicio Algoritmo GRASP Construcción Proyectos (N, V, P, D, S, α)

1. Leer archivo (input.csv)
2. Inicializar N, V, α
3. Inicializar matrices (P, D)
4. **Para i:1 a <condición de parada>**

Inicio

4.1 Costo Total= 0

4.2 S= ϕ

4.3 Mientras Costo_Total \leq V hacer
Inicio

4.3.1 C = ϕ

4.3.2 A = ϕ

4.3.3 $\beta = \text{Max} (B_i / (C_i * T_i * R_i) \forall P_i \notin S)$

4.3.4 $\tau = \text{Min} (B_i / (C_i * T_i * R_i) \forall P_i \notin S)$

4.3.5 RCL = $\{ P_i \notin S \mid \beta - \alpha * (\beta - \tau) \leq B_i / (C_i * T_i * R_i \leq \beta \}$

4.3.6 C = Aleatorio(RCL)

4.3.7 A = Analiza Dependencia(C)

4.3.8 Si A $\neq \phi$ entonces
Inicio

4.3.8.1 Costo Total = Costo Total + C_i

4.3.8.2 S = S U {A}

4.3.8.3 P = P - {A}

Fin
Fin Mientras 4.3
Fin Para 4

5. Retornar S, Costo Total

Fin Algoritmo GRASP Construcción Proyectos

Tabla 17 Pseudocódigo GRASP construcción

Para este algoritmo se puede precisar:

- Líneas 1 a 3: se leen los elementos involucradas en el problema desde un archivo llamado input.csv y se inicializan las variables N (número de proyectos), V (presupuesto), P (matriz de proyectos), D (dependencia de proyectos) y la constante α de relajación.
- Línea 4: la condición de parada es un número determinado de iteraciones a calibrar.
 - Líneas 4.1 a 4.2: se inicializan el costo total del portafolio (Costo_Total) y el conjunto solución o portafolio de proyectos (S).
 - Línea 4.3: el final de una iteración se dará en el momento en que el costo total de los proyectos seleccionados exceda el presupuesto V o se llegue a un número determinado de iteraciones.
 - Línea 4.3.1 a 4.3.2: se inicializan en vacío el valor del proyecto elegido C y el arreglo auxiliar A . Este último contendrá el proyecto seleccionado y las dependencias que ser incluidas en el portafolio.
 - Líneas 4.3.3 a 4.3.5: se obtienen los valores de β (mejor proyecto) y τ (peor proyecto) sobre la base de información de los proyectos no seleccionados hasta ese momento, formando de este modo la lista RCL.
 - Línea 4.3.6: se selecciona de manera aleatoria un proyecto de la lista RCL.
 - Línea 4.3.7: se analiza dependencia de modo que si no excede el presupuesto disponible se agregan los proyectos al arreglo auxiliar A , incluyendo el proyecto seleccionado C . Caso contrario, A es nulo.

- Línea 4.3.8: si el arreglo auxiliar A es diferente de nulo:
 - ✓ Línea 4.3.8.1 a 4.3.8.3: se añade la lista de proyectos al conjunto solución S y se agrega el costo a Costo_Total. Finalmente, se retira el proyecto o proyectos seleccionados del arreglo P que contiene proyectos aún no seleccionados para el portafolio final.
- Línea 5: se obtiene la lista que representa el portafolio de proyectos seleccionados y el costo total de dicho portafolio.



CAPÍTULO 7

1. Pseudocódigo de algoritmo Búsqueda Tabú

Tener en consideración las siguientes variables:

VARIABLE	REPRESENTACIÓN
Cantidad de proyectos	N
Presupuesto para proyectos	V
Riesgo estimado del proyecto i	R_i

Tabla 18 Variables para algoritmo Búsqueda Tabú

Adicionalmente, considerar los arreglos:

ARREGLO	REPRESENTACIÓN
Matriz de Proyectos	P
Matriz de Dependencia	D
Matriz Tabú (Reciente)	R
Matriz Tabú (Frecuente)	F
Arreglo Solución	S

Tabla 19 Estructuras de datos para algoritmo Búsqueda Tabú

A continuación se presenta el pseudocódigo del algoritmo de Búsqueda Tabú:

Inicio Algoritmo Búsqueda Tabú Proyectos (N, V, P, D, S, R, F)

1. Inicializar matrices (R, F)
2. Mejor Solución = S
3. PAux = P
- /*Fase búsqueda tabú básica*/**
- 4. Mientras no cumpla <condición de parada>**
Inicio
 4.1 Búsqueda (P, Mejor Solución, S, PAux, D, R, F)
Fin Mientras

/*Fase de intensificación*/
5. Reiniciar Estados (R, S, PAux, Mejor Solución, P)
- 6. Mientras no cumpla <condición de parada>**
Inicio
 6.1 Búsqueda (P, Mejor Solución, S, PAux, D, R, F)
Fin Mientras

/*Fase de diversificación*/
7. Reiniciar Estados (R, S, PAux, Mejor Solución, P)
8. Penalizar movimientos frecuentes (F, R)
- 9. Mientras no cumpla <condición de parada>**
Inicio
 9.1 Búsqueda (P, Mejor Solución, S, PAux, D, R, F)
- 10. Retornar Mejor Solución, Costo Total**

Fin Algoritmo Búsqueda Tabú Proyectos

Tabla 20 Pseudocódigo Búsqueda Tabú

Inicio Búsqueda (P, Mejor Solución, S, PAux, D, R, F)

1. Vec=Identificar Vecinos(S, PAux, D)
2. M = Selecciona Mejor Movimiento (Vec, R)
3. S = Realiza Mejor Movimiento (S, M, PAux, D)
4. Actualiza Listas Tabú (R, F, M)
5. Si Mejor Solución < S entonces

Inicio

5.1 Mejor Solución = S

5.2 P = PAux

Fin
Fin Búsqueda

Tabla 21 Procedimiento Búsqueda (Búsqueda Tabú)

Inicio Identificar Vecinos (S, PAux, D)

1. Vec= ϕ
2. Para i:1 a <Número de elementos de PAux>

Inicio

2.1 Para j:1 a <Número de elementos de S>

2.2 Inicio

2.2.1 Vec=Vec + Intercambio(PAux[i], S[j],D)

Fin
Fin

3. Retornar Vec

Fin Identificar Vecinos

Tabla 22 Procedimiento Identificar Vecinos (Búsqueda Tabú)

Inicio Selecciona Mejor Movimiento (Vec, R)

1. $M = \emptyset$
2. Vec= Ordenar Descendente (Vec)
3. Index=0
4. **Mientras <no se encuentre mejor movimiento>**

Inicio

- 4.1 Index = Index + 1
- 4.2 $M = \text{Vec}[\text{Index}]$
- 4.3 Si M <no es Tabú> entonces <salir Mientras>
- 4.4 Caso contrario
 - 4.4.1 Si M <cumple Criterio Aspiración> entonces <salir Mientras>

Fin

5. Retornar M

Fin Selecciona Mejor Movimiento

Tabla 23 Procedimiento Selecciona Mejor Movimiento (Búsqueda Tabú)

Para este algoritmo se puede precisar:

- **Algoritmo Búsqueda Tabú Proyectos:**
 - Línea 1: se inicializan en 0 las listas tabú de lo reciente (R) y lo frecuente (F).
 - Línea 2 a 3: se guarda la mejor solución obtenida del algoritmo GRASP construcción S y los proyectos disponibles o no seleccionados P.

/*Fase de Búsqueda Tabú Básica*/

- Línea 4: se ejecuta la fase de búsqueda tabú básica hasta que se cumpla condición de parada. La condición de parada puede venir dada por un número de iteraciones o hasta que se cumplan un número de iteraciones sin mejorar el acumulado de la función objetivo.
 - Línea 4.1: se ejecuta el procedimiento de Búsqueda.

/*Fase de Intensificación: para la fase de intensificación se requiere explorar el espacio de soluciones a partir de la mejor solución encontrada, por ello se debe reiniciar la matriz de lo reciente (R) y empezar la búsqueda desde allí*/

- Línea 5: se reinicia la lista tabú de lo reciente (R), es decir, se ponen todo los valores de la lista 0, lo que quiere decir que ningún movimiento es tabú o tiene penalización de turnos. Se asigna la mejor solución a S y los proyectos que estaban disponibles en ese momento a PAux.
- Línea 6: se ejecuta la fase de intensificación hasta que se cumpla condición de parada. La condición de parada puede venir dada por un número de iteraciones o hasta que se cumplan un número de iteraciones sin mejorar el acumulado de la función objetivo.
 - Línea 6.1: se ejecuta el procedimiento de Búsqueda.

/*Fase de Diversificación: para la fase de diversificación se requiere explorar espacios de soluciones no visitados anteriormente, por ello se debe penalizar fuertemente los movimientos frecuentes, ubicados en la lista tabú de lo frecuente (F), con la finalidad de que la solución se pueda mover del óptimo local y se reinicia la lista tabú de lo frecuente (F) para empezar nuevamente la búsqueda */

- Línea 7: se reinicia la lista tabú de lo reciente (R), es decir, se ponen todo los valores de la lista 0, lo que quiere decir que ningún movimiento es tabú o tiene penalización de turnos. Se asigna la mejor solución a S y los proyectos que estaban disponibles en ese momento a PAux.
 - Línea 8: utilizando la lista tabú de lo frecuente (F), se penalizan fuertemente los movimientos de mayor frecuencia. Esto se verá reflejado en la lista tabú de lo reciente (R) ya que estos movimientos no podrán ser ejecutados al comienzo de la búsqueda sino hasta que cumplan los turnos de penalización. Por ejemplo, se podría penalizar aquellos movimientos que han sido usados más de 10 veces.
 - Línea 9: se ejecuta la fase de diversificación hasta que se cumpla condición de parada. La condición de parada puede venir dada por un número de iteraciones o hasta que se cumplan un número de iteraciones sin mejorar el acumulado de la función objetivo.
 - Línea 9.1: se ejecuta el procedimiento de Búsqueda.
 - Línea 10: se obtiene la lista que representa el portafolio de proyectos seleccionados y el costo total de dicho portafolio.
- **Procedimiento de Búsqueda:**
 - Línea 1: se identifican las posibles soluciones a las que se puede llegar a partir de la solución actual ejecutando la función Identificar Vecinos.

- Línea 2: se selecciona el mejor movimiento no tabú de entre las posibles soluciones a ser seleccionadas. En esta selección se debe tener también en cuenta el criterio de aspiración, el cual permitirá seleccionar un movimiento tabú siempre y cuando la función obtenida sea mejor que la actual. Se obtiene ejecutando la función Selecciona Mejor Movimiento.
 - Línea 3: se realiza el movimiento seleccionado en el paso anterior.
 - Línea 4: se actualizan las listas tabú de lo reciente (R) y lo frecuente (F). En la lista de lo reciente (R) se penaliza con un número X, en los cuales este movimiento no podrá ser usado. En la lista de lo frecuente (F) se incrementa en uno la cantidad de movimientos que se ha realizado dicho movimiento.
 - Línea 5: si la función objetivo de la nueva solución S es mayor que la función objetivo de la Mejor Solución, se coloca como Mejor Solución a S y se asigna a P (proyectos disponibles) el valor de PAux.
- ***Función Identificar Vecinos:***
 - Línea 1: se inicializa la lista Vec, que contendrá las posibles soluciones a las que se llega a partir de la solución actual.
 - Línea 2: Se recorrerán las listas de proyectos disponibles PAux y la lista de solución actual S. En cada iteración se almacenará cada intercambio entre un proyecto disponible y un elemento o más de la solución actual, dependiendo si el intercambio involucra o no una dependencia de proyectos. Dicha operación se realiza en el procedimiento Intercambio, y a su vez se calcula y registra el nuevo valor acumulado de la función objetivo que tendría la nueva solución por cada intercambio a realizar.
 - Línea 3: se retorna la lista Vec que contiene los intercambios y la función objetivo que tendría la nueva solución por cada intercambio a realizar.

- **Función Selecciona Mejor Movimiento:**

- Línea 1: se inicializa M, que contendrá el mejor movimiento posible a realizar.
- Línea 2: se ordena la lista Vec, que contiene las posibles soluciones a las que se puede llegar a partir de la solución actual. El ordenamiento es de forma descendente por función objetivo, de modo que el mayor es aquel cuyo intercambio es la mejor opción. Sin embargo, no será necesariamente este el movimiento elegido, ya que se debe comprobar si es o no tabú.
- Línea 3: se inicializa Index en 0. Se utilizará para recorrer la lista Vec ordenada descendientemente.
- Línea 4: se realizan las iteraciones hasta que se encuentre el mejor movimiento. Una vez se encuentre dicho movimiento, se terminará el bucle con la instrucción <salir Mientras>.
 - Línea 4.1: se incrementa en 1 cada vez para poder recorrer los elementos de Vec.
 - Línea 4.2: se asigna a M el elemento de Vec [Index].
 - Línea 4.3: se verifica que el elemento asignado a M no se tabú, para ello se utiliza la lista tabú de lo reciente (R). En caso no sea tabú, se ejecuta la instrucción <salir Mientras>.
Caso contrario, tenemos que el elemento asignado a M es tabú. Se verifica si cumple el Criterio de Aspiración, en caso cumpla se ejecuta la instrucción <salir Mientras>.
Si es que no se cumple ninguna de las condiciones anteriores se continúa con el bucle hasta encontrar un elemento adecuado.
- Línea 5: se retorna el mejor movimiento M.

CAPÍTULO 8

1. Experimentación Numérica

A continuación se muestran los resultados de la experimentación numérica, haciendo uso de la Comparación de dos Tratamientos. Dicha prueba se desarrolla haciendo uso de los resultados obtenidos por los algoritmos GRASP construcción y Búsqueda Tabú.

Estos algoritmos fueron ejecutados con 40 veces y cada uno con una muestra de datos diferentes, es decir, 40 muestras. A continuación se muestran los resultados obtenidos:



GRASP		TABU SEARCH	
Parámetro	FO	Parámetro	FO
1	5.32	1	5.33
2	4.15	2	4.76
3	7.20	3	7.85
4	4.71	4	5.17
5	3.91	5	5.18
6	5.93	6	6.42
7	5.92	7	6.31
8	5.89	8	6.39
9	5.33	9	6.04
10	5.81	10	6.78
11	5.01	11	5.80
12	5.76	12	6.09
13	4.98	13	5.31
14	5.33	14	5.66
15	3.59	15	4.25
16	5.09	16	5.76
17	5.32	17	5.85
18	3.52	18	3.92
19	5.68	19	6.25
20	4.55	20	5.16
21	4.27	21	4.95
22	5.38	22	5.92
23	6.56	23	6.76
24	5.68	24	6.46
25	5.56	25	5.99
26	3.66	26	4.39
27	5.38	27	6.47
28	5.57	28	6.19
29	3.79	29	4.03
30	5.90	30	6.36
31	4.72	31	5.00
32	5.62	32	6.07
33	5.28	33	5.77
34	6.66	34	6.90
34	4.58	35	4.83
36	6.03	36	6.43
37	4.96	37	5.30
38	4.48	38	5.00
39	5.60	39	6.15
40	4.82	40	5.29

Ilustración 5 Muestra de datos obtenidos

1.1 Prueba Kolmogorov Smirnov

Se requiere demostrar que los resultados obtenidos mediante los algoritmos desarrollados presentan una distribución normal, para así poder efectuar los test siguientes (Prueba F y Prueba Z).

- Algoritmo GRASP construcción

Media	5.19	FO					
Desviación	0.848903319						
Mínimo	3.52						
Máximo	7.20				Estadístico S-K	0.1097	
Rango	3.68				Nivel de significancia	0.05	
Numero de Datos (N)	40				Grados de libertad	40	
# intervalos (Sturges)	13.1733022				Estadístico de la tabla	0.21012	
# intervalos (raiz de N)	6.32455532				HIPÓTESIS	Se ACEPTA	
Tamaño del intervalo	0.581691366						
Intervalos	Lím. Inferior	Lím Superior	Fo	For	For acum	Fer acum	Abs (For acum-Fer acum)
1	3.52	4.10	5	0.1250	0.1250	0.0997	0.0253
2	4.10	4.68	5	0.1250	0.2500	0.2749	0.0249
3	4.68	5.26	7	0.1750	0.4250	0.5347	0.1097
4	5.26	5.84	15	0.3750	0.8000	0.7800	0.0200
5	5.84	6.43	5	0.1250	0.9250	0.9275	0.0025
6	6.43	7.01	2	0.0500	0.9750	0.9839	0.0089
7	7.01	7.59	1	0.0250	1.0000	0.9977	0.0023
			40				

Ilustración 6 Resultado Kolmogorov-Smirnov (GRASP construcción)

- Algoritmo Búsqueda Tabú

Media	5.71	FO					
Desviación	0.8411136						
Mínimo	3.92						
Máximo	7.85				Estadístico S-K	0.058274	
Rango	3.93				Nivel de significancia	0.05	
Numero de Datos (N)	40				Grados de libertad	40	
# intervalos (Sturges)	13.1733022				Estadístico de la tabla	0.21012	
# intervalos (raiz de N)	6.32455532				HIPÓTESIS	Se ACEPTA	
Tamaño del intervalo	0.62179446						
Intervalos	Lím. Inferior	Lím Superior	Fo	For	For acum	Fer acum	Abs (For acum-Fer acum)
1	3.92	4.54	4	0.1000	0.1000	0.0815	0.0185
2	4.54	5.16	5	0.1250	0.2250	0.2560	0.0310
3	5.16	5.78	10	0.2500	0.4750	0.5333	0.0583
4	5.78	6.41	13	0.3250	0.8000	0.7947	0.0053
5	6.41	7.03	7	0.1750	0.9750	0.9409	0.0341
6	7.03	7.65	0	0.0000	0.9750	0.9893	0.0143
7	7.65	8.27	1	0.0250	1.0000	0.9988	0.0012
			40				

Ilustración 7 Resultado prueba Kolmogorov-Smirnov (Búsqueda Tabú)

Cada cuadro estadístico corresponde tanto al algoritmo GRASP construcción como al algoritmo de Búsqueda Tabú respectivamente.

Siendo las hipótesis para ambos algoritmos:

H0: El algoritmo GRASP presenta una Distribución Normal.

H1: El algoritmo GRASP no presenta una Distribución Normal.

H0: El algoritmo Búsqueda Tabú presenta una Distribución Normal.

H1: El algoritmo Búsqueda Tabú no presenta una Distribución Normal.

Esto quiere decir que si el valor estadístico S-K es menor al valor estadístico de tabla (Smirnov - Kolmogorov) para 40 muestras, entonces la solución está fuera de la región crítica, por lo que se aceptaría la hipótesis H0, afirmando que corresponde a una Distribución Normal. Caso contrario, si el valor estadístico S-K es mayor o igual al valor estadístico de tabla.

Como se puede observar, en ambos casos el valor estadístico S-K obtenido es menor que el valor estadístico de tabla para 40 muestras, por lo que se puede afirmar que la larga y en promedio no existe evidencia estadística contraria para rechazar el modelo propuesto, el cual pretende demostrar que ambos resultados presentan una distribución normal.

1.2 Prueba F de Fischer

Una vez demostrada la normalidad de las soluciones mediante la prueba de Kolmogorov – Smirnov, es necesario garantizar que dichas soluciones también poseen varianzas homogéneas, con el fin de posteriormente efectuar la Prueba Z, la cual permite comparar las medias poblacionales y así demostrar estadísticamente que los resultados de un algoritmo son mejor que los resultados del otro.

Hipótesis		Conclusión	
H ₀ : Las varianzas son significativamente homogéneas		Varianzas HOMOGÉNEAS	
H ₁ : Las varianzas son significativamente diferentes			
		GRASP	TABU SEARCH
Media	5.19	5.71	
Varianza	0.720636844	0.70747	
Observaciones	40	40	
Grados de libertad	39	39	
F	1.0186081575		
P(F<=f) una cola (derecha)	0.4771922448		
Valor crítico para F (una cola)	1.7044650671		

Ilustración 8 Resultados prueba F de Fischer

Las hipótesis a evaluar son:

H₀: Las varianzas son significativamente homogéneas

H₁: Las varianzas son significativamente diferentes

Esto quiere decir que si el valor estadístico de F obtenido es menor que el valor crítico para F (una cola), entonces la solución está fuera de la región crítica, por lo tanto se aceptaría H₀, afirmando que las varianzas son significativamente homogéneas. Caso contrario si el valor estadístico de F es mayor o igual que el valor crítico para F (una cola).

Como se puede observar en el cuadro, el valor resultante de la prueba (F) es menor que el valor crítico de F para dicha prueba, por lo tanto se acepta la hipótesis de que ambas soluciones poseen varianzas similares.

1.3 Prueba Z

Ya demostrado que los datos arrojados por los algoritmos presentan una distribución normal y poseen varianzas homogéneas o semejantes, se procede a efectuar la comparación de medias. Ya que se están analizando 40 muestras, se utiliza la Prueba Z, en caso las muestras fueron menores de 35 se hubiera optado por la prueba T-Student.

Comparación 1
H0: La media de GRASP es igual que la media de TABU SEARCH
H1: La media de GRASP es diferente que la media de TABU SEARCH

Conclusión 1
Medias de GRASP son DIFERENTES que TABU SEARCH

Comparación 2
H0: La media de GRASP es igual que la media de TABU SEARCH
H1: La media de GRASP es menor que la media de TABU SEARCH

Conclusión 3
Medias de GRASP es MENOR que TABU SEARCH

	GRASP	TABU SEARCH
Media	5.19	5.71
Varianza	0.720636844	0.707472092
Observaciones	40	40
z	-2.78281	
P(Z<=z) una cola	0.00269	
Valor crítico de z (una cola)	1.64485	
P(Z<=z) dos colas	0.00539	
Valor crítico de z (dos colas)	1.95996	

Ilustración 9 Resultados prueba Z

Para esta prueba se presentan dos comparaciones:

- Comparación 1: esta primera comparación consiste en demostrar que los resultados de ambos algoritmos poseen medias diferentes, para lo cual se recurre a la prueba de dos colas utilizando como hipótesis:

H0: La media de GRASP es igual que la media Búsqueda Tabú

H1: La media de GRASP es diferente que la media de Búsqueda Tabú

Para ello se usa el valor en tabla de Valor crítico de z (dos colas), dicho esto, para aceptar como cierta la hipótesis nula el valor de Z debe estar entre - 1.95996 y +1.95996, sin embargo observamos que dicho valor es -2.78281 por lo que caería en una región crítica haciendo que se rechace la hipótesis nula y llegar a la conclusión que tanto las medias del algoritmo GRASP construcción y Búsqueda Tabú son diferentes.

- Comparación 2: esta segunda comparación consiste en demostrar que la media del resultado del algoritmo GRASP construcción es mayor que la media del algoritmo Búsqueda Tabú, para lo cual se recurre a la prueba de una cola utilizando como hipótesis:

H0: La media de GRASP es mayor que la media Búsqueda Tabú

H1: La media de GRASP es menor que la media de Búsqueda Tabú

Para ello se usa el valor en tabla de Valor crítico de z (una cola).

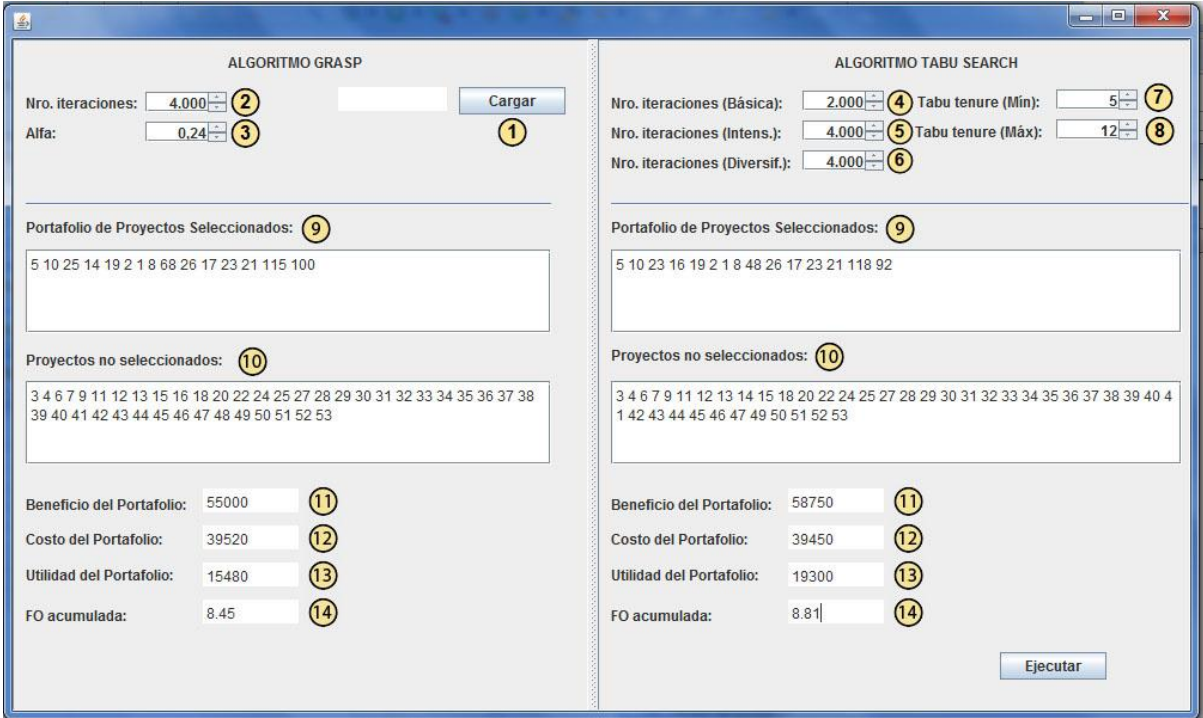
Para aceptar como cierta la hipótesis nula el valor de Z debe ser mayor a -1.64485 , sin embargo observamos que dicho valor es -2.78281 por lo que caería en una región crítica haciendo que se rechace la hipótesis nula y llegar a la conclusión que la media del GRASP construcción es menor a la media de Búsqueda Tabú.



CAPÍTULO 9

1. Interfase

A continuación se muestra la interfaz en la que se presentarán los resultados de ejecutar tanto el algoritmo GRASP construcción como el algoritmo de Búsqueda Tabú.



The screenshot shows a software interface with two main columns. The left column is titled 'ALGORITMO GRASP' and the right column is titled 'ALGORITMO TABU SEARCH'. Both columns have similar input fields and sections. In the GRASP column, 'Nro. iteraciones' is 4,000 (2), 'Alfa' is 0,24 (3), and there is a 'Cargar' button (1). The 'Portafolio de Proyectos Seleccionados' (9) contains the numbers 5 10 25 14 19 2 1 8 68 26 17 23 21 115 100. The 'Proyectos no seleccionados' (10) contains the numbers 3 4 6 7 9 11 12 13 15 16 18 20 22 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53. Summary statistics for GRASP are: Beneficio del Portafolio: 55000 (11), Costo del Portafolio: 39520 (12), Utilidad del Portafolio: 15480 (13), FO acumulada: 8.45 (14). The Tabu Search column has 'Nro. iteraciones (Básica): 2,000 (4), Tabu tenure (Min): 5 (7), Nro. iteraciones (Intens.): 4,000 (5), Tabu tenure (Máx): 12 (8), and Nro. iteraciones (Diversif.): 4,000 (6). The 'Portafolio de Proyectos Seleccionados' (9) contains the numbers 5 10 23 16 19 2 1 8 48 26 17 23 21 118 92. The 'Proyectos no seleccionados' (10) contains the numbers 3 4 6 7 9 11 12 13 14 15 18 20 22 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 49 50 51 52 53. Summary statistics for Tabu Search are: Beneficio del Portafolio: 58750 (11), Costo del Portafolio: 39450 (12), Utilidad del Portafolio: 19300 (13), FO acumulada: 8.81 (14). A 'Ejecutar' button is at the bottom right.

Ilustración 10 Interfaz para comparación de algoritmos

- 1: Se debe ingresar un archivo creado mediante el generador de datos aleatorios.
- 2: Se puede configurar el número de iteraciones en que se ejecutará el algoritmo GRASP construcción.
- 3: Se puede configurar la constante de relajación alfa. Debe ser un número entre 0 y 1.
- 4: Se puede configurar el número de iteraciones de la Fase Básica de la Búsqueda Tabú.

- 5: Se puede configurar el número de iteraciones de la Fase de Intensificación de la Búsqueda Tabú.
- 6: Se puede configurar el número de iteraciones de la Fase de Diversificación de la Búsqueda Tabú.
- 7: Se puede configurar el número mínimo de tenencia tabú
- 8: Se puede configurar el número máximo de tenencia tabú
- 9: Se muestran los proyectos del portafolio seleccionado.
- 10: Se muestran los proyectos que no fueron seleccionados para el portafolio.
- 11: Se muestra el beneficio que se obtendría con el portafolio de proyectos seleccionados.
- 12: Se muestra el costo que tendría el portafolio de proyectos seleccionados.
- 13: Se muestra la utilidad que tendría el portafolio de proyectos seleccionados.
- 14: Se muestra la función objetivo acumulada como resultado del portafolio de proyectos seleccionado

CAPÍTULO 10

1. Conclusiones

Para finalizar el presente proyecto de fin de carrera, el cual incluye los algoritmos GRASP construcción y algoritmo de Búsqueda Tabú, para resolver el problema de Selección de Proyectos de TI se puede concluir lo siguiente:

El algoritmo GRASP construcción tuvo que ser calibrado, de modo que se pueda obtener mejores soluciones. Esto debido a que tiene un factor de aleatoriedad que si no se controla adecuadamente puede arrojar soluciones no tan buenas. Para el caso abordado se obtuvo como resultado de la calibración el valor de 0.24, cabe también mencionar que el número de iteraciones para este algoritmo es de 32000, ya que es el número recomendado según bibliografía.

Por otro lado el algoritmo de Búsqueda Tabú cuenta con 3 fases, las cuales son Búsqueda Básica, Intensificación y Diversificación, esto con la finalidad de que la solución no se entrampe en un óptimo local. Para cada una de estas fases se definió un número de iteraciones, las cuales son de 2000, 4000 y 4000, respectivamente. Se estableció de este modo ya que según bibliografía deben ser al menos 1000, sin embargo esto varía de problema a problema, por lo que se están tomando números mayores al recomendado.

Otro factor a tener en cuenta es el Tabu Tenure o Tenencia Tabú, el cual es el número de iteraciones que estará prohibido un intercambio para lograr una solución, que para el caso particular de este proyecto se hizo de forma dinámica, es decir, no es un número fijo sino que se establece un intervalo del cual se selecciona un valor aleatoriamente. Este factor es dinámico ya que según bibliografía se obtienen mejores resultados que si fuera estático.

Ambos algoritmos se llevaron a comparación de modo que se pueda demostrar que el algoritmo de Búsqueda Tabú es mejor en cuanto a soluciones obtenidas. Para ello se elaboró un experimento numérico, comparación de dos tratamientos, que afirmó dicha teoría.

En cuanto a resultados obtenidos se puede observar que no necesariamente el obtener mayor utilidad significa que se deba tomar ese portafolio, sino que hay otros factores de riesgo que deben ser evaluados y tomados en cuenta. Por tanto, se puede decir que se obtiene un portafolio rentable y, si bien no está libre de riesgos de pérdida de dinero, existe una probabilidad bastante alta que este sea exitoso.

Además se debe mencionar que ambos algoritmos son configurables, puesto que se pueden modificar el número de iteraciones, la constante de relajación alfa para el GRASP construcción y el Tabu Tenure para Búsqueda Tabú. Para el caso de este último algoritmo pueden requerirse más iteraciones, dependiendo del número de proyectos a evaluar.

Finalmente se puede decir que el presente proyecto de fin de carrera cumple con los objetivos planteados, ya que se obtiene un portafolio de proyectos rentable y haciendo una evaluación de menor tiempo a si se evaluara cada opción por separado como se realiza actualmente en las empresas. Por tanto, podría ser usado para la toma de decisiones sobre qué proyectos de entre tantos otros son los más adecuados para el portafolio.

2. Recomendaciones y trabajos futuros

Se propone como trabajos futuros a partir del presente proyecto de fin de carrera:

- La adaptación del algoritmo propuesto a la variante de que en lugar de realizar la selección de proyectos para un portafolio, este sea para múltiples portafolios, es decir, multiportafolio.
- Modificar la función objetivo, de modo que el algoritmo pueda adaptarse a otros tipos de proyectos, como por ejemplo proyectos de construcción donde una variable adicional puede ser la zonificación donde se construirá.
- Implementar un sistema, el cual incluya el algoritmo propuesto, de modo que sea parte de la toma de decisiones en cuestión de selección de proyectos.

Referencias bibliográficas

AUGEO SOLUTIONS

2012 <http://www.augeo.com/page/managing-business-projects>

BECK Kent y Cynthia Andres

2008 Extreme Programming Explained: Embrace Change. Second Edition

BERENSON Mark, David Levine y Timothy Krehbiel

2006 Estadística para Administración. Pearson Educación.

BIN, Qu

2009 “An Immune Algorithm for Optional Selection Problem of InvestmentProjects”. Fifth International Conference on Natural Computation. China

CARAZO, Ana, Trinidad GÓMEZ y Fátima PÉREZ

2011 “Análisis de los principales aspectos que afectan a la decisión de selección y planificación de carteras de proyectos”. Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA. Málaga, 2011, Volumen 12. Páginas 123 – 140.

CUEVA, TUPIA, GUANIRA

2013 “A Genetic Algorithm to solve the IT investment project selection”.

DICKINSON, Michael, Anna Thornton y Stephen Graves

2001 “Technology Portfolio Management: Optimizing Interdependent Projects over Multiple Time Periods”. IEEE Transactions On Engineering Management. Volumen 48. Páginas 518-527.

DONG, Cheng

2010 “Risk Analysis of Investment Projects”. International Conference on E-Business and E-Government. Hubei province, China.

FEO, Thomas y Mauricio RESENDE

1995 “Greedy Randomized Adaptive Search Procedures”. Journal of Global Optimization. Páginas 109 - 134.

FULGA, Cristinca y Silvia DEDU

2010 “A New Approach in Multi-Objective Portfolio Optimization using Value-at-Risk based Risk Measure”. IEEE. Páginas 765-769.

GALLART, Joseph

2009 Análisis, diseño e implementación de un algoritmo metaheurístico GRASP que permita resolver el problema de rutas de vehículos con capacidad/Tesis para optar por el título de Ingeniero Informático. Lima: Pontificia Universidad Católica del Perú.

GALLART Joseph y otros

2010 “Generación Inteligente de Horarios empleando heurísticas GRASP con Búsqueda Tabú para la Pontificia Universidad Católica del Perú”. Revista de Ingeniería Informática PUCP. Lima, 2010, Volumen 1, Páginas 15-24.

GAN Yunfeng y Xiaowei GAO

2011 "A Research on Multiple Projects Selection Base on Matching Matrix Model".IEEE. Páginas 84-87.

GANOZA Dante y Ursula Solano

2004 "Un algoritmo de búsqueda adaptativa aleatoria y golosa para la resolución del problema de cortes"/Tesis para optar por el Título Profesional de: INGENIERO DE SISTEMAS E INFORMÁTICA. Lima: Universidad Nacional Mayor de San Marcos.

GLOVER, Fred

1989 "Tabu Search".Journal on Computing.Volumen 3.

GLOVER, Fred y Belén MELIÁN

2003 "TabuSearch". Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Volumen 19. Páginas 29-48.

GLOVER, Fred y Manuel LAGUNA.

Primera Edición.

USA: KluwerAcademicPublishers 1997

INTERNATIONAL SOCIETY ON MULTIPLE CRITERIA DECISION MAKING

2013 <http://www.mcdmsociety.org/>

JAVA

2013 <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>

LI-HUA Tian y Li Wen-hua

2011 "The Research of The Financial Evaluation Methods of Investment Projects Based on EVA". IEEE. Páginas 368-371.

MARTELLO, Silvano y Paolo TOTH. Knapsack Problem.

Primera Edición.

1990

MARTI, Rafael

2003 "Procedimientos Metaheurísticos en Optimización Combinatoria". Matemáticas. Páginas 1-60.

MARTINEZ Alejandro y Raúl Martínez, "Guía a Rational Unified Process"

<http://www.docstoc.com/docs/120443927/Gu%C3%ADa-completa-RUP>

MELIÁN, Belén y Fred GLOVER

2004 "Introducción a la búsqueda Tabú". Universidad de La Laguna

MORENO José

2004 Metaheurísticas: Conceptos y Propiedades. Universidad de La Laguna. Tenerife: España.

MORENO Pilar y otros

2007 "Metaheurísticas de Optimización Combinatoria: Uso de Simulated Annealing para un problema de calendarización". Tecnología y Desarrollo. Madrid, 2007, Volumen 5.

NASA, Computing, Information, and Communications Technology Program
2013, [http://ti.arc.nasa.gov/m/pub-archive/697h/0697%20\(Alfano\).pdf](http://ti.arc.nasa.gov/m/pub-archive/697h/0697%20(Alfano).pdf)

NETBEANS

2013 <https://netbeans.org/features/index.html#>

PAPADIMITRIOU, Christos y Kenneth STEIGLITZ. Combinatorial Optimization:
Algorithms and Complexity
Prentice Hall, 1998.

PMBOK Guía de los Fundamentos para la Dirección de Proyectos (Guía del
PMBOK) PROJECT MANAGEMENT INSTITUTE
2009, Cuarta Edición. Pennsylvania: PMI Publications.

PROJECT MANAGEMENT OFFICE

2013 <http://www.projectportfoliooffice.com/ppo.php?page=1/>

PROJECT PORTFOLIO OPTIMIZATION

2013 <http://www.pwc.com/us/en/audit-assurance-services/valuation/fund-portfolio-capital-projects.jhtml#projectsum>

RETURETA, Elsa

2010 Estadística Inferencial: Planteamiento de Hipótesis en más de dos
poblaciones. Universidad Veracruzana: Facultad de Administración.

SOREN, Eilertsen

2004 "The Art of Project Selection".Kolner Group

TUPIA, Manuel.

2009. Fundamentos de Inteligencia Artificial Primera Edición. Lima: Tupia Consultores y Auditores S.A.C.

TUPIA, Manuel

2005 “Un algoritmo GRASP para resolver el problema de la programación de tareas dependientes en máquinas diferentes”/Tesis para optar por el título de Magister en Ingeniería de Sistemas con la mención en Ingeniería de Software. Lima: Universidad Nacional Mayor de San Marcos

WANG, Yanzhang y Shengju HAN

1997 “A multiple criteria 0-1 linear programming method of project planning and investment decision making”. Innovation in Technology Management - The Key to Global Leadership. PICMET '97: Portland International Conference on Management and Technology. China.