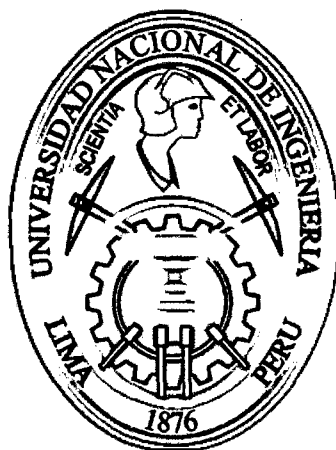


UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

SECCION DE POSGRADO



**“SOFTWARE DE DISTRIBUCION HORARIA DE CLASES
APLICANDO EL METODO HEURISTICO TABU SEARCH”**

TESIS

**PARA OPTAR EL GRADO ACADEMICO DE MAESTRO
CIENCIAS CON MENCIÓN EN:
INGENIERIA DE SISTEMAS**

Ing. Miguel Wilhelm Ruiz Túpac Yupanqui

LIMA - PERU

Digitalizado por:

2012

Consortio Digital del
Conocimiento MebLatam,
Hemisferio y Dalse

A mis Padres por haber logrado con su apoyo constante el poder llegar hasta aquí.

A mi esposa, por su ayuda, paciencia y comprensión.

A mis hijos Cristina, Jorge y Astrid por ser el motor motivador de todos mis esfuerzos...

INDICE

CARÁTULA.....	1
INDICE.....	2
RESUMEN.....	4
ABSTRACT.....	5
PALABRAS CLAVES.....	6
INTRODUCCION.....	7
CAPITULO I PLANTEAMIENTO DEL PROBLEMA.....	1
1.1 DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA.....	1
1.2 DEFINICION DEL PROBLEMA.....	3
1.3 DELIMITACION DE LOS OBJETIVOS.....	4
1.3.1 OBJETIVO GENERAL.....	4
1.3.2 OBJETIVOS ESPECIFICOS.....	4
1.4 HIPÓTESIS DE LA INVESTIGACION.....	4
1.4.1 <i>HIPÓTESIS</i>	4
1.4.2 IDENTIFICACION Y OPERACIONALIZACION DE VARIABLES.....	5
1.5 MATRIZ DE CONSISTENCIA.....	7
1.6 JUSTIFICACION Y DELIMITACION DE LA INVESTIGACION.....	8
1.6.1 IMPORTANCIA DEL TEMA.....	8
1.6.2 JUSTIFICACION.....	9
1.6.3 DELIMITACION Y ALCANCE DEL TRABAJO.....	9
CAPITULO II MARCO TEORICO Y CONCEPTUAL.....	10
1.7 ESTADO DEL ARTE Y TRABAJOS PREVIOS.....	10
1.8 MARCO TEORICO.....	11
1.8.1 META HEURISTICAS.....	11
1.8.2 META HEURISTICAS APLICABLES.....	16
1.8.3 NORMAS PARA EL DESARROLLO DE HORARIOS DE EDUCACION SECUNDARIA.....	30

1.9	MARCO CONCEPTUAL.....	34
1.9.1	SUSTENTO TEORICO DEL PROYECTO.....	34
CAPITULO III. MÉTODO HEURISTICO TABU SEARCH.....		39
1.10	INTRODUCCION.....	39
1.10.1	EL PROBLEMA DE LA GENERACION DE HORARIOS.....	39
1.10.2	Función objetivo.....	40
1.10.3	Movimientos posibles.....	41
1.10.4	REQUERIMIENTOS Y LIMITACIONES.....	42
1.10.5	FORMULACIÓN MATEMÁTICA (pseudocódigo).....	43
1.11	APLICACIÓN DEL MÉTODO HEURISTICO TABU SEARCH EN LA GENERACION DE HORARIOS.....	45
1.11.1	Entidades:.....	45
1.11.2	ALGORITMO DEL MÉTODO DE TABU SEARCH PARA LA PROGRAMACIÓN DE HORARIOS (PSEUDOCÓDIGO).....	51
CAPITULO IV ANALISIS DE LOS RESULTADOS.....		56
1.12	PRESENTACION E INTERPRETACION DE RESULTADOS.....	56
1.13	CONTRASTACION DE HIPOTESIS.....	70
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES.....		74
CONCLUSIONES.....		74
RECOMENDACIONES.....		76
GLOSARIO DE TERMINOS.....		78
BIBLIOGRAFIA.....		80
ANEXOS.....		84

RESUMEN

El presente trabajo está basado en la aplicación de la metaheurística Tabu Search en la distribución horaria de colegios secundarios estatales peruanos con el fin de poder mejorar el rendimiento académico de los alumnos.

Hay estudios que muestran que el rendimiento académico de los estudiantes no es el mismo a cada hora del día, especialmente cuando se trata de los cursos de mayor complejidad cognoscitiva (matemáticas, ciencias, etc.).

Al poder aplicar una distribución horaria acorde con estas expectativas se puede incrementar el aprendizaje de los estudiantes.

ABSTRACT

This work is based on the application of the Tabu Search metaheuristic in the Timetabling problem of Peruvian secondary schools in order to improve the academic performance of students.

Studies show that students' academic performance is not the same at different times of day, especially when it comes to more complex cognitive courses (math, science and so on).

Being able to apply a time distribution in line with these expectations can increase student learning.

PALABRAS CLAVES

Horarios Escolares, Secundaria, Perú, Pedagógico, meta heurísticas, Tabu Search, métodos probabilísticos, generación automática, métodos aleatorios, métodos guiados.

INTRODUCCION

Las meta heurísticas son métodos probabilísticos que han sido aplicadas con éxito en problemas para los cuales no existe un algoritmo que encuentre su solución óptima en tiempo razonable: los problemas NP-Completos. Un problema de esta categoría es la generación de horarios escolares, el cual generalmente es un proceso tedioso y consumidor de recursos de diversa índole, especialmente de horas-hombre.

Evaluar las diferentes técnicas que influyen positivamente en la generación de horarios escolares, y desarrollar un algoritmo específico para los horarios escolares peruanos (estatales). Se considera un fin pedagógico poder alcanzar cerca de 10 objetivos entre los cuales los más importantes son completar todas las horas en todas las secciones y obtener la mayor cantidad de horas dobles en las primeras horas de enseñanza para los cursos de ciencias, evitar los cursos de ciencias los días viernes, etc.

Si bien la asignación de horarios escolares (School Timetable Problem) es considerado un problema combinatorial clásico, su alta dependencia del sistema educacional lo convierte en NP Completo, justificando en muchos de los casos el empleo de heurísticas.

Todo el desarrollo estará basado haciendo un estudio técnicas Heurísticas Tabú Search.

El desarrollo del trabajo está estructurado de la siguiente forma

En el **Capítulo I**: Planteamiento del Problema, Enunciado, Delimitación, Objetivos, Hipótesis, etc.

En el **Capítulo II** Marco Teórico y Conceptual, Métodos Heurísticos Aplicables

En el **Capítulo III** Método Heurístico Tabu Search, Algoritmo tabu search y formulación de un algoritmo mejorado de tabu search.

En el **Capítulo IV** Análisis de los Resultados, aplicación del algoritmo creado y análisis de resultados.

En el **Capítulo V** Conclusiones y Recomendaciones

En el **Anexo I** Tenemos los diferentes valores obtenidos por la función costo al aplicar el algoritmo.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA

La calidad de un horario, incide significativamente en el rendimiento académico del alumno. Calidad de un horario lo entendemos como que cada horario en cada sección cumpla con ciertos requisitos en la distribución de cursos a fin de los alumnos puedan asimilar mejor los conocimientos.

Se ha hecho estudios que cuando los alumnos reciben los cursos de ciencias 2 horas seguidas las primeras horas de clase, les permite asimilar con mayor eficiencia los conocimientos.

Así mismo los cursos de ciencias cuando se les aplica mayor carga los días viernes, tienden a ser menos asimilados por los alumnos.

Del mismo modo si los cursos de ciencias se imparten la última hora estos no son asimilados con la misma eficiencia.

Las secciones que no presentan estas características tienden a tener un menor rendimiento académico en general.

Por otro lado la generación de Automática de Horarios, conocido

como Timetabling es un problema NP- difícil, requiere de una alta complejidad, tiempos de procesamiento extensivo en algunos casos de alrededor de 10^{21} años usando la tecnología computacional actual.

Lo más usual es realizar la generación de horarios de manera manual lo que conlleva a dedicarle semanas e incluso meses de trabajo, sin poder lograr un resultado que satisfaga a todos los requerimientos deseables. Debido a esto se limita en cierta forma los resultados pedagógicos.

Por este motivo se requiere de hacer uso de la meta heurísticas y diseñar un algoritmo que haga viable una generación automática exitosa. Si bien estas heurísticas no nos proporcionaran una solución óptima, nos dan una solución adecuada.

Existen heurísticas definidas para tratar de resolver este problema tales como sistemas expertos, búsqueda tabú, algoritmos genéticos, etc.

En el presente trabajo se abordara las técnica heurística Tabú Search para resolver el problema específico de poder generar horarios escolares peruanos, que tienen ciertas características particulares como por ejemplo que toda una sección o grupo de alumnos llevan los mismos cursos con los mismos profesores, no existen cursos electivos, un buen horario estará dado cuando se pueda completar la asignación de horas a toda una institución educativa. No hay cambios de cursos entre profesores, es decir antes de empezar el problema ya se tienen definidos estos parámetros.

En nuestro país la generación de horarios en los colegios estatales de secundaria se hace de manera manual, donde un encargado llena los horarios en la medida de lo posible cumpliendo las restricciones obligatorias, pero difícilmente al no contar con un sistema automatizado puede mejorar este horario a fin de dar una mayor calidad a las asignaciones horarias cumpliendo restricciones sugeridas, tales como que los cursos de ciencias de dicten en las primeras horas, que exista en buen balance de cursos a lo

largo de la semana y entre las secciones , es decir que todas las secciones tengan una calidad de horario similares.

Es decir al ser realizados los horarios de una manera manual se limita bastante la calidad en la generación del mismo, y por consiguiente afecta la calidad de la enseñanza en los colegios, limita a su vez la disponibilidad horaria de los profesores, de recursos de aulas, consumiendo muchas horas hombre en el encargado de hacer los mismos.

En los colegios particulares, en algunos pocos, se usan sistemas automatizados, pero al provenir de otros países de una realidad diferente, no consiguen el grado de satisfacción que desearían. En los demás se hace al igual que los colegios estatales de manera manual.

De tener un sistema automatizado eficiente que se adapte a nuestra realidad, mejoraría significativamente la calidad de la distribución horaria tanto para los alumnos, profesores, administrativos y por consiguiente de la institución, por lo cual los alumnos mejoran significativamente su aprendizaje.

1.2 DEFINICION DEL PROBLEMA

¿En qué medida el modelamiento de un software basado en el método heurístico tabu search genera una distribución horaria que mejora el rendimiento académico del alumno de un colegio del estado de nivel secundario?

¿En qué medida se puede uniformizar la asignación horaria de todas las secciones de cada grado, a fin de que tengan ventajas similares en la distribución horaria?

¿En qué medida se pueden seleccionar diferentes tipos de restricciones a fin de obtener una mejor eficiencia en la distribución horaria?

1.3 DELIMITACION DE LOS OBJETIVOS

1.3.1 OBJETIVO GENERAL

Modelar un software de distribución horaria aplicando el método heurístico tabu search que mejore el rendimiento académico del alumno de un colegio del estado de nivel secundario

1.3.2 OBJETIVOS ESPECIFICOS

Uniformizar la asignación horaria de todas las secciones de cada grado con la finalidad que tengan ventajas similares en la distribución horaria.

Determinar los tipos de restricciones que mejoren la eficiencia en la distribución horaria.

1.4 HIPÓTESIS DE LA INVESTIGACION

1.4.1 HIPÓTESIS

1.4.1.1 HIPOTESIS GENERAL

El modelamiento de un software de distribución horaria basado en el método heurístico tabu search mejora el rendimiento académico del alumno de un colegio del estado de nivel secundario.

1.4.1.2 HIPOTESIS ESPECÍFICAS

El uniformizar la asignación horaria de todas las secciones de cada grado, permite que tengan ventajas similares de distribución horaria.

La Selección diferentes tipos de restricciones permitirán obtener una mejor eficiencia en la distribución horaria.

1.4.2 IDENTIFICACION Y OPERACIONALIZACION DE VARIABLES

DEFINICION CONCEPTUAL DE LAS VARIABLES

Modelamiento de un Software (Variable Interviniente):

Conjunto de Técnicas específicas para obtener un Producto de Software necesario.

Método heurística Tabu Search (Variable Interviniente):

Heurística que permite resolver problemas de alta complejidad (NP Difícil).

Distribución Horaria (Variable Independiente):

Matriz de distribución Académica donde se define los cursos a dictarse durante el día por cada Sección.

Rendimiento Académico (Variable Dependiente):

Eficiencia de lo aprendido por los alumnos.

DEFINICION OPERACIONAL DE LAS VARIABLES

Tipo de Modelamiento de un Software:

Basándose en indicadores de calidad tales como nivel de alineamiento y eficiencia se define un software que optimice en gran medida los resultados esperados, observando:

Alineamiento: número de objetivos logrados,

Eficiencia: Memoria consumida vs cursos asignados.

Método heurístico Tabu Search:

Permite un menor tiempo de generación horaria, se seleccionara creando el algoritmo que demande menos consumo de memoria.

Capacidad: Memoria consumida vs cursos asignados.

Distribución Horaria:

Matriz obtenida con el modelamiento de software siguiendo un conjunto de indicadores. Dicha distribución se buscara primero minimizando los traslapes, y luego maximizando los objetivos definidos.

Numero de traslapes: Numero de cruces vs numero de horas totales.

Rendimiento Académico:

Depende de la distribución horaria para ser optimizado. Se puede obtener haciendo un estudio comparativo midiendo el promedio de aula por sección y por promoción a lo largo del tiempo.

1.5 MATRIZ DE CONSISTENCIA

Problema	Objetivos	Hipótesis	Variables	Indicadores	Índice
Problema principal ¿En qué medida el modelamiento de un software basado en el método heurístico tabu search genera una distribución horaria que mejora el rendimiento académico del alumno de un colegio del estado de nivel secundario?	Objetivo general Modelar un software de distribución horaria aplicando el método heurístico tabu search que mejore el rendimiento académico del alumno de un colegio del estado de nivel secundario	Hipótesis General El modelamiento de un software de distribución horaria basado en el método heurístico tabu search mejorará el rendimiento académico del alumno de un colegio del estado de nivel secundario.	Tipo de Modelamiento de un Software: Método heurístico Tabu Search. Distribución Horaria: Rendimiento Académico.	Alineamiento Eficiencia Capacidad Nivel de traslape Promedio General Aula	Numero de objetivos alineados vs objetivos totales Tiempo de procesamiento vs núm. cursos asignados. Memoria consumida vs cursos asignados Número de cruces vs número de horas totales Promedio Total de cada promedio por alumno.
Problema específico Nro1 ¿En qué medida se pueden seleccionar diferentes tipos de restricciones a fin de obtener una mejor eficiencia en la distribución horaria?	Objetivo específico Nro1. Determinar los tipos de restricciones que mejoren la eficiencia en la distribución horaria.	Hipótesis Específica Nro1. La Selección diferentes tipos de restricciones permitirán obtener una mejor eficiencia en la distribución horaria.	Restricciones Distribución Horaria:	Restricciones Aplicables Calidad por restricciones	Numero de restricciones $Calidad = \text{Producto de Vector de Calidad} * \text{Vector de Ponderación}$
Problema específico Nro2 ¿En qué medida se puede uniformizar la asignación horaria de todas las secciones de cada grado, a fin de que tengan ventajas similares en la distribución horaria?	Objetivo específico Nro2. Uniformizar la asignación horaria de todas las secciones de cada grado con la finalidad que tengan ventajas similares en la distribución horaria.	Hipótesis Específica Nro2. El uniformizar la asignación horaria de todas las secciones de cada grado, permite que tengan ventajas similares de distribución horaria.	Uniformidad de Asignaciones Horarias	Nivel de Dispersión Horaria	$\text{Índice de Uniformidad} = \text{Calidad de Horario Max} / \text{Calidad de Horario Min.}$

1.6 JUSTIFICACION Y DELIMITACION DE LA INVESTIGACION

1.6.1 IMPORTANCIA DEL TEMA.

Toma auge una teoría que muestra cómo el cerebro tiene momentos más adecuados que otros para asimilar una materia. Los colegios empiezan a aplicar estas ideas en sus horarios porque no todas las actividades cognitivas se pueden hacer en cualquier momento del día.

A la luz de nuevas investigaciones esto se explica porque no todas las actividades cognitivas se pueden hacer en cualquier momento del día. Existe una hora ideal en ese ciclo de 24 horas para asimilar el conocimiento, otra para memorizar, otra para jugar y otra para dormir

Anteriormente se establecían los horarios de clase sin ninguna racionalidad. Se tenía la idea de que mientras más temprano fuera, mejor se realizarían las actividades que demandaban una mayor complejidad, como las matemáticas.

Pero con el desarrollo de la cronopsicología, una ciencia que busca entender cuál es el ritmo del cerebro durante el día para optimizar el aprendizaje de los niños, ha quedado claro que hay una diferencia del cielo a la tierra si se dicta esa asignatura a las 8 o a las 10 de la mañana.

Una correcta distribución horaria donde se priorice que los cursos con mayor demanda de complejidad cognoscitiva a cierta banda horaria y ciertos días de la semana pueden provocar que los estudiantes asimilen los conocimientos con una mayor eficiencia.

1.6.2 JUSTIFICACION

Este problema tiene fundamental importancia, debido a que correctamente aplicado y realizando una distribución horaria más eficiente podría aumentar la eficiencia de la calidad en la enseñanza educativa de cientos de colegios estatales del Perú. Además de mejorar la disponibilidad horaria de los profesores del país. Y por añadidura disminuir el tiempo administrativo que se dedica a la generación manual de horarios y los errores que ello implique.

1.6.3 DELIMITACION Y ALCANCE DEL TRABAJO

El desarrollo de las actividades llevará a cabo en un Colegio de Educación Secundaria apoyo del área de Coordinación Académica de la Institución, y con apoyo de los actores involucrados en el proceso de generación de horarios.

CAPITULO II

MARCO TEORICO Y CONCEPTUAL

1.7 ESTADO DEL ARTE Y TRABAJOS PREVIOS

En el Perú a nivel de colegios secundarios, los horarios de clase se han venido realizando de manera manual utilizando pocos criterios de optimización, debido a la propia naturaleza limitada al realizarlo de esta forma.

En algunos colegios particulares se han realizado a través de programas importados que muchas veces no cumplen con las características de distribución horaria en nuestra realidad, por lo cual termina realizándose el trabajo también de forma manual.

En los tiempos actuales aun se sigue realizando la distribución horaria de manera manual en los colegios estatales en los colegios estatales, sin poder aprovechar de un sistema automatizado que simplifique el proceso y que pueda realizar una distribución que permita utilizar nuevos criterios a fin de poder mejorar el aprendizaje del alumnado.

Existen sin embargo algunos estudios en otros países de aplicación de

metaheurísticas para la distribución horaria, pero estas no son optimizadas para los colegios del Perú donde poseen características diferentes y ni aplican criterios de optimización según estos parámetros, solo lo aplican a fin de poder completar un horario y lograr criterios mínimos de optimización.

MOLINA ARAYA, Juan (2007)

Universidad de Valparaíso (Chile)

Se centró en encontrar una solución al problema de asignación de horarios que existe en las carreras de Ingeniería en Informática Aplicada e Ingeniería Civil Informática de la Universidad de Valparaíso usando Algoritmos Evolutivos (AE).

- Haroldo G. Santos, Luiz S. Ochi, and Marcone J.F. Souza (2005)

Universidad Federal Fluminense (Brasil)

Empleo un algoritmo de Tabu Search utilizando estrategias de diversificación. Teniendo en cuenta la historia del proceso de búsqueda para guiar la Selección de los movimientos de diversificación obtuvo resultados favorables en la solución de problemas de horarios.

1.8 MARCO TEORICO

1.8.1 META HEURISTICAS

Los problemas de optimización que implican un gran número finito de alternativas surgen con frecuencia en la industria y la ciencia. En estos problemas, existe un conjunto finito de soluciones, X , y una función real, $f : X \rightarrow \mathbb{R}$, y se busca una solución $x^* \in X$, con $f(x^*) \leq f(x), \forall x \in X$. Para encontrar la solución óptima a un problema de optimización es posible, teóricamente, enumerar las soluciones y evaluar cada una de ellas con respecto al objetivo marcado. Sin embargo, desde un punto de vista práctico, es imposible seguir esta estrategia de resolución, porque el número de combinaciones

frecuentemente crece de forma exponencial conforme al tamaño del problema.

Muchos de los trabajos que se han elaborado durante las últimas cinco décadas para el desarrollo de métodos de optimización que no requieran una evaluación exhaustiva de todos los electos del campo de soluciones. Esta búsqueda ha afectado fundamentalmente al campo de la optimización combinatoria y a la capacidad de solucionar problemas complejos de la vida real. Sin embargo, muchos problemas que aparecen en la industria son computacionalmente intratables por su naturaleza, o porque son tan complejos que imposibilitan el uso de algoritmos exactos. En estos casos, los métodos heurísticos se emplean habitualmente para encontrar buenas soluciones, pero no necesariamente garantizan que sean las óptimas. La efectividad de estos métodos depende de su capacidad de adaptarse a cada caso particular y la capacidad para evitar quedar atrapados en óptimos locales. Teniendo en cuenta estas nociones, se ha demostrado que las técnicas heurísticas permiten obtener buenas soluciones a complejos problemas de optimización combinatoria

1.8.1.1 Procedimientos heurísticos

De acuerdo con ANSI/IEEE Std 100-1984 (American National Standards Institute / Institute of Electrical and Electronic Engineers), la heurística trata de métodos o algoritmos exploratorios durante la resolución de problemas complejos en los cuales las soluciones se descubren por la evaluación del progreso logrado en la búsqueda de un resultado final. No se puede garantizar que dichas soluciones sean las óptimas, pero sí razonablemente buenas.

Así pues, la heurística es una serie de procedimientos o estrategias que se supone nos llevan a un destino deseado; pero esto no

tiene por qué ser siempre cierto. Cuando queremos ir hacia el este, podemos establecer un sistema tal como esperar que salga el sol y caminar en la dirección del nacimiento del sol. Lamentablemente esta estrategia no nos garantiza que podamos esquivar o sortear la presencia de obstáculos durante el camino: muros, ríos, montañas u otros elementos.

Los algoritmos heurísticos suelen ser métodos de resolución de problemas muy útiles cuando se cumplen una o múltiples de las siguientes circunstancias:

- Cuando el problema o la situación planteada no requieren una solución exacta o precisa, sino sólo aproximada.
- Cuando los datos disponibles son poco fiables.
- Cuando no puede aplicarse un método exacto (simplex, branch & bound, etc.) para la resolución del problema, por no existir o porque requiera demasiados recursos (tiempo de cálculo, memoria, presupuesto, etc.).
- Como un paso intermedio para la aplicación de otros algoritmos. Muchas veces la solución aportada por un método heurístico es tomada como punto de partida de otros procedimientos (especialmente metaheurísticos).
- Cuando las limitaciones de tiempo, espacio (para el almacenaje de datos), etc. conducen a la utilización de métodos de respuesta rápida aunque sea a costa de la precisión.

El término heurística proviene de la palabra griega "heuriskein" relacionado con el concepto de encontrar. Se califica de heurístico a un procedimiento que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad. Las heurísticas son métodos para resolver problemas complejos en forma aproximada.

Además, una ventaja importante es que su complejidad es reducida en comparación con los métodos exactos, por lo que suelen ser

más fáciles de entender (por parte de los directivos de las empresas y gente no experta). Son flexibles y tienen como objetivo encontrar soluciones de buena calidad en un tiempo computacional razonable sin mencionar que generalmente ofrecen más de una solución, permitiendo así ampliar las posibilidades de elección.

A pesar de sus ventajas, no cabe duda de que cuando una técnica exacta está disponible debe gozar de preferencia sobre cualquier tipo de heurística, sobre todo cuando los valores económicos manejados sean importantes y el tiempo para resolverlos no esté limitado.

En los últimos años ha habido un crecimiento en el desarrollo de procedimientos heurísticos para resolver problemas combinatorios, debido a la necesidad de disponer de herramientas que permitan ofrecer soluciones rápidas a problemas reales. Los problemas de optimización combinatoria en particular son de gran dificultad debido a su complejidad, ya que crecen exponencialmente con el tamaño del problema, por lo cual se pretende que los métodos heurísticos se acerquen a la solución óptima en un tiempo razonable.

1.8.1.2 Clasificación de los métodos heurísticos

Existen diferentes tipos de heurísticas, las cuales pueden clasificarse como:

a) **Métodos constructivos:** Son aquellos que añaden componentes individuales a una solución parcial hasta que se obtiene una solución factible. El más popular de estos métodos lo constituye un algoritmo goloso o voraz, "greedy", el cual construye la solución buscando el máximo beneficio en cada paso. Un ejemplo es el GRASP.

b) **Métodos de descomposición:** Consisten en dividir el problema en subproblemas más pequeños, siendo la salida de uno la entrada de otro, de

forma que al resolver ambos subproblemas obtengamos una solución para el problema global. Un ejemplo de aplicación en un problema de programación lineal mixta, consistiría en decidir de alguna forma la solución para las variables enteras para luego resolver el problema como un programa lineal.

c) Métodos de reducción: Identifican alguna característica que deba poseer la Solución óptima y de este modo simplifican el problema. Por ejemplo la detección de alguna variable con ciertos valores o correlación.

d) Métodos de manipulación del modelo: Modifican las estructuras del modelo con el fin de hacerlo más sencillo de resolver, deduciendo, a partir de la solución del problema modificado, la solución del problema original. Como por ejemplo, se puede reducir el espacio de soluciones o eliminando restricciones del problema.

e) Métodos de búsqueda por entornos: Parten de una solución factible inicial

(Probablemente obtenida de otra heurística) y mediante alteraciones de la solución, van iterando a otras factibles de su entorno, almacenando la mejor solución encontrada hasta que se cumpla un determinado criterio de parada.

1.8.1.3 Procedimientos metaheurísticos

Una metaheurística es un método heurístico para solucionar una clase muy general de problemas de cómputo combinando los procedimientos de caja negra –generalmente heurísticos- de una manera eficiente. El nombre combina el prefijo griego “meta” (“más allá de”, aquí en el sentido de “a un nivel superior”) y “heurístico”

La metaheurística se aplica generalmente a los problemas para

los cuales no hay algoritmo específico satisfactorio o heurístico para su resolución; o cuando no es práctico poner tal método en ejecución. La mayoría de meta heurísticas de uso general se aplica sobre problemas de optimización combinatoria puros o todos aquéllos que puedan adaptarse a esa naturaleza, como por ejemplo solucionar ecuaciones booleanas.

Según Osman y Kelly (1995), una definición podría ser la siguiente: "Los procedimientos meta heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Las meta heurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y mecanismos estadísticos".

1.8.2 META HEURISTICAS APLICABLES

1.8.2.1 Recocido Simulado (SA, Simulated Annealing)

La aproximación SA fue desarrollada por Kirkpatrick, Gelatt y Vecchi (1983) y se le conoce como Recocido Simulado dado la analogía entre la simulación del recocido de sólidos y el problema de la resolución de los grandes problemas de optimización combinatoria.

SA parte de una solución inicial determinada por una heurística, y mediante la exploración de su entorno, trata de encontrar la solución óptima. El método consiste de iteraciones, donde se compara la solución en curso con la solución vecina generada aleatoriamente. Si la solución vecina es mejor que la solución en curso, entonces la solución vecina pase a ser la

solución en curso.

En el caso de que la solución vecina sea peor que la solución en curso se puede tomar esta solución vecina como la solución en curso, con una probabilidad que depende de la diferencia entre la solución vecina y la solución en curso y la temperatura T. La regla de generación de la probabilidad de aceptación de la solución vecina que es peor que la solución en curso sigue la distribución Boltzmann: Donde la función tomará valores entre [0,1], y dependerá de la solución vecina, la solución en curso, la constante de Boltzmann k y la temperatura T en la iteración n.

Layout of Simulated Annealing

Input: A problem instance

Output: A (sub-optimal) solution

1. Generate an initial solution at random and initialize the temperature T.
2. **While** (T>0) **do**
 - a. **While** (thermal equilibrium not reached) **do**
 - i. Generate a neighbor state at random and evaluate the change in energy level $\Delta E > 0$.
 - ii. If $\Delta E < 0$ update current state with new state.
 - iii. If $\Delta E \geq 0$ update current state with new state with probability $e^{\frac{-\Delta E}{k_B T}}$.
 - b. Decrease temperature T according to annealing schedule.
3. Output the solution having the lowest energy.

Figura 1. Esquema de Algoritmo Recocido Simulado...[16]

1.8.2.2 GRASP (Greedy Randomized Adaptive Search Procedure)

La técnica de esta metaheurística de tipo iterativo, desarrollada originalmente por Feo y Resende (1989) al estudiar un problema de cobertura de alta complejidad combinatoria, se basa en la combinación de fases de explotación, en las cuales se siguen algoritmos de búsqueda local, con fases de exploración, en los que se usan procedimientos aleatorios para expandir el espacio de búsqueda recorrido.

El algoritmo GRASP mantiene una lista de posibles soluciones a un problema, que se genera aleatoriamente. Se escoge una de las soluciones, y se mejora hasta que no se pueda mejorar más; si es mejor que la mejor almacenada, se actualiza la lista de soluciones almacenadas.

Cada iteración en GRASP consta generalmente de dos pasos: la fase de construcción y el procedimiento de búsqueda local. En el primero se construye una solución tentativa, que luego es mejorada mediante un procedimiento de intercambio hasta que se llega a un óptimo local. En la fase constructiva, GRASP toma en cuenta la función objetivo con la intención de que al término de la iteración se cuente con una solución de alta calidad, sobre la cual se efectúa una mejora o Fase II.

Layout of GRASP

Input: A problem instance

Output: A (sub-optimal) solution

1. Initialization: set $x^* = \infty$.
2. **While** (stopping criterion not satisfied) **do**
 - a. Construct a greedy randomized solution x ;
 - b. Find local minimum \tilde{x} in neighborhood $N(x)$ of x ;
 - c. If $f(\tilde{x}) < f(x^*)$ then update $x^* = \tilde{x}$;
3. Output the best solution found x^* .

Figura 2. Esquema de Algoritmo GRASP...[16]

Layout of GRASP Construction Phase

Input: A problem instance and pseudo random number stream.

Output: A (sub-optimal) solution

1. Initialization: set solution $S = 0$;
2. **While** (solution construction not done) **do**
 - a. Using greedy function, make restricted candidate list (RCL);
 - b. At random, select element s from RCL;
 - c. Place s in solution i.e. $S = S \cup \{s\}$;
 - d. Change greedy function to take into account update S ;
3. Output the solution x corresponding to set S ;

Figura 3. Esquema de Algoritmos GRASP: fase construcción...[16]

1.8.2.3 Algoritmo Genético (GA, Genetic Algorithm)

El GA es un algoritmo de búsqueda que explora un espacio de solución que simula procesos en un sistema natural hacia la evolución, específicamente aquellos que siguen el principio de la supervivencia en función de la adaptabilidad (Adenso Díaz et al, 1996). Fue desarrollado por Holland (1975), y se distingue muy claramente de todos los anteriores, básicamente por

El hecho de que en cada iteración se tiene un conjunto de soluciones, o población en curso y no una única solución en curso. Las soluciones sucesoras se obtienen a partir de parejas constituidas con los elementos de la población y no mediante la transformación de la solución en curso.

El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad alrededor del mundo durante los últimos años. Se presentarán aquí los conceptos básicos que se requieren para abordarla, así como un sencillo ejemplo que permita a los lectores comprender cómo aplicarla al problema de su elección. Adicionalmente, se hablará acerca de los diversos ambientes de programación actuales basados en algoritmos genéticos y de las áreas abiertas de investigación.

Orígenes

En los últimos años, la comunidad científica internacional ha mostrado un creciente interés en una nueva técnica de búsqueda basada en la teoría de la evolución y que se conoce como el algoritmo genético. Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes (unidad básica de codificación de cada uno de los atributos de un ser vivo) de un

individuo, y que los atributos más deseables (i.e., los que le permiten a un individuo adaptarse mejor a su entorno) del mismo se transmiten a sus descendientes, cuando éste se reproduce sexualmente.

Un investigador de la Universidad de Michigan llamado John Holland estaba consciente de la importancia de la selección natural, y a fines de los 60s desarrolló una técnica que permitió incorporarla en un programa de computadora. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo 1975.

Cuando Holland se enfrentó a los algoritmos genéticos, los objetivos de su investigación fueron dos:

- Imitar los procesos adaptativos de los sistemas naturales
- Diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Una definición bastante completa de un algoritmo genético es:

Es un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

¿Cómo saber si es posible usar el Algoritmo Genético?

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y

confiables. Sin embargo, no todos los problemas pudieran ser apropiados para la técnica, y se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla:

- Su espacio de búsqueda (i.e., sus posibles soluciones) debe estar delimitado dentro de un cierto rango.
- Debe poderse definir una función de aptitud que nos indique qué tan buena o mala es una cierta respuesta.
- Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

El primer punto es muy importante, y lo más recomendable es intentar resolver problemas que tengan espacios de búsqueda discretos -aunque éstos sean muy grandes-. Sin embargo, también podrá intentarse usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando exista un rango de soluciones relativamente pequeño.

La función de aptitud no es más que la función objetivo de nuestro problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente utilizando el recíproco de la función maximizante (debe cuidarse, por supuesto, que el recíproco de la función no genere una división por cero). Una característica que debe tener esta función es que debe ser capaz de "castigar" a las malas soluciones, y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez.

La codificación más común de las respuestas es a través de cadenas binarias, aunque se han utilizado también números reales y letras. El primero de estos esquemas ha gozado de mucha popularidad debido a que es el que propuso originalmente Holland, y además porque resulta muy sencillo de

implementar.

Funcionamiento de un Algoritmo Genético Simple

La operación de un algoritmo genético simple puede ilustrarse con el generar población inicial, $G(0)$;

evaluar $G(0)$;

$t:=0$;

Repetir

$t:=t+1$;

Generar $G(t)$ usando $G(t-1)$;

Evaluar $G(t)$;

Hasta encontrar una solución;

Primero, se genera aleatoriamente la población inicial, que estará constituida por un conjunto de cromosomas, o cadenas de caracteres que representan las soluciones posibles del problema. A cada uno de los cromosomas de esta población se le aplicará la función de aptitud a fin de saber qué tan buena es la solución que está codificando.

Sabiendo la aptitud de cada cromosoma, se procede a la selección de los que se cruzarán en la siguiente generación (presumiblemente, se escogerá a los "mejores"). Dos son los métodos de selección más comunes:

es muy simple, y consiste en crear una ruleta en la que cada cromosoma

tiene asignada una fracción proporcional a su aptitud. Sin que nos refiramos a una función de aptitud en particular, supongamos que se tiene una población de 5 cromosomas cuyas aptitudes están dadas por los valores mostrados en la Tabla 1.

Cromosoma No.	1	2	3	4	5	Total
Cadena	11010110	10100111	00110110	01110010	11110010	
Aptitud	254	47	457	194	85	1037
% del Total	24.5	4.5	44.1	18.7	8.2	100

Tabla 1: Valores de ejemplo para ilustrar la selección mediante ruleta

Con los porcentajes mostrados en la cuarta columna de la Tabla 1 podemos elaborar la ruleta de la Figura 1. Esta ruleta se gira 5 veces para determinar qué individuos se seleccionarán. Debido a que a los individuos más aptos se les asignó un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos.

2) El torneo: La idea de este método es muy simple. Se baraja la población y después se hace competir a los cromosomas que la integran en grupos de tamaño predefinido (normalmente compiten en parejas) en un torneo del que resultarán ganadores aquéllos que tengan valores de aptitud más altos. Si se efectúa un torneo binario (i.e., competencia por parejas), entonces la población se debe barajar 2 veces. Nótese que esta técnica garantiza la obtención de múltiples copias del mejor individuo entre los progenitores de la siguiente generación (si se efectúa un torneo binario, el mejor individuo será seleccionado 2 veces).

Una vez realizada la selección, se procede a la reproducción sexual o cruce de los individuos seleccionados. En esta etapa, los sobrevivientes

intercambiarán material cromosómico y sus descendientes formarán la población de la siguiente generación. Las 2 formas más comunes de reproducción sexual son: uso de un punto único de cruza y uso de 2 puntos de cruza.

Cuando se usa un solo punto de cruza, éste se escoge de forma aleatoria sobre la longitud de la cadena que representa el cromosoma, y a partir de él se realiza el intercambio de material de los 2 individuos.

Cuando se usan 2 puntos de cruza, se procede de manera similar, pero en este caso el intercambio se realiza en la forma mostrada en la Figura 3.

Normalmente la cruza se maneja dentro de la implementación del algoritmo genético como un porcentaje que indica con qué frecuencia se efectuará. Esto significa que no todas las parejas de cromosomas se cruzarán, sino que habrá algunas que pasarán intactas a la siguiente generación. De hecho existe una técnica desarrollada hace algunos años en la que el individuo más apto a lo largo de las distintas generaciones no se cruza con nadie, y se mantiene intacto hasta que surge otro individuo mejor que él, que lo desplazará. Dicha técnica es llamada elitismo, y no debe sorprendernos el hecho de que haya sido desarrollada en Alemania.

Además de la selección y la cruza, existe otro operador llamado mutación, el cual realiza un cambio a uno de los genes de un cromosoma elegido aleatoriamente. Cuando se usa una representación binaria, el gene seleccionado se sustituye por su complemento (un cero cambia en uno y viceversa). Este operador permite la introducción de nuevo material cromosómico en la población, tal y como sucede con sus equivalentes biológicos.

Al igual que la cruza, la mutación se maneja como un porcentaje que indica con qué frecuencia se efectuará, aunque se distingue de la primera por ocurrir mucho más esporádicamente (el porcentaje de cruza normalmente es

de más del 60%, mientras que el de mutación normalmente nunca supera el 5%).

Si supiéramos la respuesta a la que debemos llegar de antemano, entonces detener el algoritmo genético sería algo trivial. Sin embargo, eso casi nunca es posible, por lo que normalmente se usan 2 criterios principales de detención: correr el algoritmo genético durante un número máximo de generaciones o detenerlo cuando la población se haya estabilizado (i.e., cuando todos o la mayoría de los individuos tengan la misma aptitud).

¿Qué Ventajas y Desventajas tienen con respecto a otras técnicas de búsqueda?

- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.
- Cuando se usan para problemas de optimización -maximizar una función objetivo- resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen -tamaño de la población, número de generaciones, etc.-.

- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

Ambientes de Programación

En la actualidad existe un gran número de ambientes de programación disponibles en el mercado para experimentar con los algoritmos distinguirse 3 clases de ambientes de programación:

1) **Sistemas Orientados a las aplicaciones:** Son esencialmente "cajas negras" para el usuario, pues ocultan todos los detalles de implementación. Sus usuarios -normalmente neófitos en el área- los utilizan para un cierto rango de aplicaciones diversas, pero no se interesan en conocer la forma en qué estos operan. Ejemplos de este tipo de sistemas son: Evolver (Axcelis, Inc.) y XpertRule GenAsys (Attar Software).

2) **Sistemas Orientados a los algoritmos:** Soportan algoritmos genéticos específicos, y suelen subdividirse en:

- **Sistemas de uso específico:** Contienen un solo algoritmo genético, y se dirigen a una aplicación en particular. Algunos ejemplos son: Escapade (Frank Hoffmeister), GAGA (Jon Crowcroft) y Genesis (John Grefenstette).
- **Bibliotecas:** Agrupan varios tipos de algoritmos genéticos, y diversos operadores (e.g. distintas formas de realizar la cruce y la selección). Evolution Machine (H. M. Voigt y J. Born) y OOGA (Lawrence Davis) constituyen 2 ejemplos representativos de este grupo.

En estos sistemas se proporciona el código fuente para que el usuario -normalmente un programador- pueda incluir el algoritmo genético en sus propias aplicaciones.

3) Cajas de Herramientas: Proporcionan muchas herramientas de programación, algoritmos y operadores genéticos que pueden aplicarse en una enorme gama de problemas. Normalmente se subdividen en:

- Sistemas Educativos: Ayudan a los usuarios novatos a introducirse de forma amigable a los conceptos de los algoritmos genéticos. GA Workbench (Mark Hughes) es un buen ejemplo de este tipo de ambiente.
- Sistemas de Propósito General: Proporcionan un conjunto de herramientas para programar cualquier algoritmo genético y desarrollar cualquier aplicación. Tal vez el sistema más conocido de este tipo es Splicer (NASA).

Áreas de Investigación

Durante los últimos años una gran parte de la investigación en esta área se ha concentrado en el desarrollo de mejoras al desempeño de los algoritmos genéticos. Se han propuesto nuevas técnicas de representación, selección y cruce, con resultados muy alentadores. Por ejemplo, el uso de los códigos de Gray y la codificación dinámica han superado algunos de los problemas asociados con la representación de valores reales mediante cadenas binarias. También se han propuesto técnicas adaptativas que varían dinámicamente los parámetros de control (porcentajes de mutación y cruce), en contraposición con el esquema estático tradicional. Otras innovaciones notables son los algoritmos genéticos distribuidos y los algoritmos genéticos paralelos. Una buena recopilación de este trabajo puede encontrarse en

Por otra parte, el fundamento matemático de los algoritmos genéticos sigue siendo otra área abierta de investigación. Resulta de especial interés el desarrollo de modelos de la dinámica de los algoritmos genéticos, el análisis de problemas que resultan difíciles para la técnica, pruebas de convergencia y en general, el tratar de comprender mejor cómo funcionan. Para los

interesados en saber qué se ha logrado

Probablemente uno de los trabajos de investigación más notables en el área sea el realizado por John Koza, que desarrolló una técnica de cruza que permite evolucionar expresiones-S de un programa en LISP. Esta innovadora forma de programación automática puede aplicarse a un enorme número de problemas, y por tal razón Koza -que por cierto fue discípulo de Holland, al igual que Goldberg- se apresuró a patentar prueba muy popular entre los expertos en computación. Su obra incluye código en Common LISP para que los interesados puedan experimentar con lo que ha dado en llamarse Programación Genética. Yendo todavía más lejos, John Koza nos muestra en un libro de más problemas de mayor complejidad, y da testimonio de implementaciones realizadas en lenguajes procedurales como C y C++, rompiendo así el mito de que sólo un lenguaje de las características propias de LISP podría utilizarse para la programación genética. La propuesta de Koza ha sido acogida con beneplácito por un gran número de investigadores alrededor del mundo, sobre todo en el área de aprendizaje máquina.

Layout of Genetic Algorithm

Input: A problem instance

Output: A (sub-optimal) solution

1. T00. Initialize P(t), and evaluate the fitness of the individuals in P(t).
2. **While** (termination condition is not satisfied) **do**
 - a. $t = t + 1$
 - b. Select P(t), recombine P(t) and evaluate P(t)
3. Output the best solution among all the population as the (sub-optimal) solution.

Figura 4. Esquema de Algoritmos Genéticos...[16]

1.8.2.4 Búsqueda Dispersa (SS, Scatter Search)

La Búsqueda Dispersa es un procedimiento metaheurístico basado en estrategias para combinar reglas de decisión, así como en la combinación de restricciones. El método SS opera sobre un conjunto de soluciones, llamado conjunto de referencia, combinando éstas para crear nuevas soluciones de modo que mejoren a las que las originaron. En este sentido decimos que es un método evolutivo.

Sin embargo, a diferencia de otros métodos evolutivos, como los algoritmos genéticos, SS no está fundamentado en la aleatorización sobre un conjunto relativamente grande de soluciones sino en elecciones sistemáticas y estratégicas sobre un conjunto pequeño, pues SS se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de reglas, restricciones o soluciones puede ser utilizado mediante la combinación de éstas en lugar de aisladamente.

1.8.3 NORMAS PARA EL DESARROLLO DE HORARIOS DE EDUCACION SECUNDARIA

DIRECTIVAS PARA EL DESARROLLO DEL AÑO ESCOLAR 2012 - MINISTERIO DE EDUCACION PERUANO

VIII.II. Educación Primaria

En el nivel de Educación Primaria, el trabajo pedagógico diario es de 6 horas pedagógicas; 30 horas semanales, y un mínimo de 1100 horas anuales. En el caso de contextos en los que sus características climatológicas y dinámicas socio culturales y económicas lo ameriten la institución educativa puede modificar la jornada escolar y el calendario siempre y cuando se cumpla con el mínimo de horas lectivas efectivas de acuerdo al numeral VII.I Las instituciones de Educación Primaria que se han registrado como

Bilingües ofrecerán progresivamente un servicio de educación intercultural bilingüe.

En el trabajo pedagógico, la organización del currículo es preferentemente por unidades didácticas integradoras. La organización de horarios de clases, se programa preferentemente en sesiones de clase con bloques de dos o más horas continuas, utilizando el tiempo de manera óptima para aprendizajes de calidad.

Las instituciones educativas participarán en actos celebratorios de la localidad o actividades propuestas por organizaciones sociales, privadas o públicas solo si contribuyen al Programa Curricular de Aula porque son parte de la diversificación curricular y forman parte del Plan Anual de Trabajo. No se contabilizan para el número mínimo de horas lectivas.

Con la finalidad de preservar el tiempo compartido, la recreación y el diálogo en las familias de asignarse tareas académicas en los fines de semana, ellas no deberán exceder de un tiempo estimado de dos horas.

Los recursos y materiales educativos que se diseñen y utilicen en los procesos de diversificación, deberán responder a la diversidad sociocultural de cada región y asegurar aprendizajes culturalmente pertinentes al contexto local, regional, nacional y universal.

Los estudiantes con prescripción médica documentada, que impida la realización de actividades físicas, pueden exonerarse, durante las ocho primeras semanas del año escolar de las actividades prácticas del Área de Educación Física a petición escrita del padre de familia o tutor; esta exoneración no, les exime de desarrollar la parte teórica del área.

Los estudiantes que profesan una religión distinta a la católica o no profesan ninguna pueden exonerarse del curso, durante las primeras ocho semanas del año lectivo, a petición escrita del padre de familia o tutor.

La IE entrega los Certificados Oficiales de Estudio oportunamente, bajo responsabilidad, a quienes hayan aprobado todos los grados del nivel.

Las y los docentes con título de profesor(a) o licenciado(a) en Educación Secundaria nombrados en Educación Primaria, deberán asumir preferentemente las secciones de 5to. ó 6to. Grado. Asimismo; las y los

docentes de Educación Inicial nombrados en Primaria deberán asumir los grados de 1er. ó 2do. Grado.

VIII.III Educación Secundaria

En el nivel secundario, el trabajo mínimo diario es de 7 horas pedagógicas, 35 horas semanales y 1200 horas anuales. En el caso de contextos en los que sus características climatológicas y dinámicas socio culturales y económicas lo ameriten la institución educativa puede modificar la jornada escolar y el calendario siempre y cuando se cumpla con el mínimo de horas lectivas efectivas de acuerdo al numeral VII.I

Las instituciones de Educación Secundaria que se han registrado como Bilingües ofrecerán progresivamente un servicio de educación intercultural bilingüe.

En la elaboración del Horario de Clases, las áreas curriculares que se desarrollan en dos horas pedagógicas, se programan obligatoriamente en bloques de horas continuas, considerando la generación de condiciones favorables para el aprendizaje, las etapas de su desarrollo evolutivo y la prevalencia del criterio superior del estudiante.

Las instituciones educativas participarán en actos celebratorios de la localidad o actividades propuestas por organizaciones sociales, privadas o públicas solo si contribuyen al Programa Curricular de Aula porque son parte de la diversificación curricular y forman parte del Plan Anual de Trabajo. No se contabilizan para el número mínimo de horas lectivas.

Con la finalidad de preservar el tiempo compartido, la recreación y el diálogo en las familias de asignarse tareas académicas en los fines de semana, ellas no deberán exceder de un tiempo estimado de dos horas.

Está prohibido el uso, la difusión y publicidad de la denominación de "Centros Pre-universitarios" para todas las IE públicas y privadas. Se prohíbe además bajo responsabilidad del Director de la institución educativa, caracterizar u ofertar la formación impartida como "pre-universitaria".

Los estudiantes con prescripción médica documentada, que impida la realización de actividades físicas, son exonerados de las actividades prácticas del Área de Educación Física. Esta exoneración no les exime de

desarrollar la parte teórica del área.

Los estudiantes que profesan una religión distinta a la católica o no profesan ninguna pueden exonerarse del curso a petición del padre de familia o tutor.

La determinación de los estudiantes, que al culminar su secundaria obtuvieron los más altos promedios generales, se realiza de acuerdo a la norma específica vigente.

Las Instituciones Educativas entregan oportunamente y bajo responsabilidad los Certificados Oficiales de Estudio, a quienes hayan aprobado todos los grados del nivel, así como los documentos de orden de méritos a quienes corresponde,

ORGANIZACIÓN Y DISTRIBUCIÓN DEL TIEMPO - EDUCACIÓN SECUNDARIA					
ÁREAS CURRICULARES	GRADOS DE ESTUDIOS				
	1°	2°	3°	4°	5°
MATEMÁTICA	4	4	4	4	4
COMUNICACIÓN	4	4	4	4	4
INGLES	2	2	2	2	2
ARTE	2	2	2	2	2
HISTORIA, GEOGRAFÍA Y ECONOMÍA	3	3	3	3	3
FORMACIÓN CIUDADANA Y CÍVICA	2	2	2	2	2
PERSONA, FAMILIA Y RELACIONES HUMANAS	2	2	2	2	2
EDUCACIÓN FÍSICA	2	2	2	2	2
EDUCACIÓN RELIGIOSA	2	2	2	2	2
CIENCIA, TECNOLOGÍA Y AMBIENTE	3	3	3	3	3
EDUCACIÓN PARA EL TRABAJO	2	2	2	2	2
TUTORÍA	1	1	1	1	1
HORAS DE LIBRE DISPONIBILIDAD	6	6	6	6	6
TOTAL DE HORAS	35	35	35	35	35

En los Centros Rurales de Formación en Alternancia (CRFA) de Educación Secundaria el Plan de Estudios se desarrolla en 50 horas pedagógicas, que incluyen 44 horas obligatorias y 6 horas de libre disponibilidad.

La IE pública que anteriormente fue colegio de variante técnica debe priorizar el área de educación para el trabajo, siempre y cuando así lo determine el diagnóstico de las necesidades pedagógicas y las IE tengan las condiciones necesarias para ello. Las familias profesionales y sus diferentes títulos y certificaciones se especifican en la RVM N° 0085-2003-ED.

En las instituciones educativas privadas este horario puede ampliarse de

acuerdo a su proyecto educativo institucional.

VIII.IV Educación Básica Alternativa

Los Centros de Educación Básica Alternativa (CEBA) organizan su atención en periodos promocionales los mismos que atienden bajo la forma presencial o la forma semi presencial.

La atención semi presencial será normada por la DIGEBA

Los Consejos de Participación Estudiantil (COPAE) participan en la gestión de los aprendizajes.

Los CEBA, atienden en el Centro Referencial o en redes periféricas en una determinada territorialidad.

La atención del PRONAMA se desarrolla a través de los CEBA

1.9 MARCO CONCEPTUAL

1.9.1 SUSTENTO TEORICO DEL PROYECTO

Búsqueda Tabú (TS, Tabu Search)

TS es un procedimiento meta heurístico de alto nivel introducido y desarrollado en su forma actual por Fred Glover (1989). Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje (Glover y Melián, 2003).

A diferencia de otros algoritmos basados en técnicas aleatorias de búsqueda de soluciones cercanas, TS es determinista ya que elimina el azar en sus decisiones y la búsqueda del óptimo está guiada por una estrategia basada en el uso de estructuras de memoria, que guardan soluciones a corto y largo plazo.

El aspecto fundamental del procedimiento consiste en comenzar

desde una solución inicial y generar un conjunto de soluciones de su vecindario. De éstas elegir la mejor de las vecinas aún cuando ésta sea peor que la solución en curso. La característica importante de TS es esencialmente la construcción de una lista tabú de movimientos: aquellos movimientos que no son permitidos (movimientos tabú) en la presente iteración. La razón de esta lista es la de excluir los movimientos que pueden hacer regresar a algún punto de una iteración anterior, es decir evitar regresar al mismo óptimo local.

Debido a que la lista tabú puede prohibir soluciones que tengan una calidad superior a la de las soluciones ya conocidas, se hizo uso del concepto "nivel de aspiración". Primero se comprueba si la solución vecina actual satisface el nivel de aspiración. Si es así, la solución vecina sustituye a la solución en curso y a la mejor. En caso contrario, si dicha solución vecina es igual o peor que la mejor, entonces se compara con los atributos de la lista tabú.

Layout of Tabu Search

Input: A problem instance

Output: A (sub-optimal) solution

1. Initialization;

- a. Generate an initial solution x and set $x^* = x$;
- b. Initialize the tabu list $T = 0$;
- c. Set iteration counters $k=0$ and $l=0$:

2. while $(N(x) \setminus T \neq \emptyset)$ do

- a. $k = k + 1$; $l = l + 1$;
- b. Select x as the best solution from set $N(x) \setminus T$;
- c. If $f(x) < f(x^*)$ the update $x = x^*$ and set $l=0$;
- d. If $k = \bar{k}$ or $l = \bar{l}$ go to step 3;

3. Output the best solution found x^* :

Figura 5. Esquema de Tabu Search...[16]

Horario Ideal para aprender

Toma auge una teoría que muestra cómo el cerebro tiene momentos más adecuados que otros para asimilar una materia. Los colegios empiezan a aplicar estas ideas en sus horarios porque no todas las actividades cognitivas se pueden hacer en cualquier momento del día.

A la luz de nuevas investigaciones esto se explica porque no todas las actividades cognitivas se pueden hacer en cualquier momento del día. Existe una hora ideal en ese ciclo de 24 horas para asimilar el conocimiento, otra para memorizar, otra para jugar y otra para dormir.

Anteriormente se establecían los horarios de clase sin ninguna racionalidad. Se tenía la idea de que mientras más temprano fuera, mejor se realizarían las actividades que demandaban una mayor complejidad, como las matemáticas.

Pero con el desarrollo de la cronopsicología, una ciencia que busca entender cuál es el ritmo del cerebro durante el día para optimizar el aprendizaje de los niños, ha quedado claro que hay una diferencia del cielo a la tierra si se dicta esa asignatura a las 8 o a las 10 de la mañana.

En Europa, por ejemplo, donde estos conceptos han sido más investigados y aplicados, en las primeras horas de la mañana los niños tienen un pequeño recreo, pues en ese momento el cerebro apenas está calentando motores.

Las clases que requieren de mayor demanda intelectual se dejan para el horario comprendido entre 10 y 11 de la mañana o a momentos intermedios de la tarde, cuando la activación del cerebro, o sea qué tan dispuesto está para recibir señales externas, llega a su punto ideal: ni demasiado ni demasiado poco. Y las tareas que tienen que ver con la motricidad se programan para las últimas horas de la jornada.

También se consideran los ritmos semanales. Los cinco días hábiles que la conforman, según Christian Hederich, uno de los investigadores colombianos que más sabe del tema, conforman un ciclo muy largo y pesado. El fin de semana también lo es. Por eso es común que en algunos colegios franceses corten ese período con una tarde libre los miércoles. El ciclo semana-fin de semana es demasiado brusco incluso para los adultos "y por eso los lunes son terribles. Cuando el cuerpo se ha acostumbrado a otro ritmo le toca empezar nuevamente", afirma Hederich.

Esta rama de la psicología busca entender mejor los ciclos de la actividad cerebral humana, su periodicidad y sus momentos de mayor y de menor eficacia. Sus conceptos están inmersos en un campo aún más amplio, el de la cronobiología, que estudia todo tipo de ritmos fisiológicos como el latir del corazón, la respiración, las funciones digestivas e incluso la menstruación. Dichos ciclos conforman una especie de reloj que trabaja con gran precisión para controlar todos los procesos dentro del organismo.

El tema de la cronobiología ha sido ampliamente aplicado en la medicina para saber a qué hora del día es mejor hacer un examen diagnóstico o recetar la toma de un medicamento. También ha sido de gran utilidad en el incremento de la productividad de las empresas, para establecer turnos de acuerdo con los cambios de atención de los trabajadores.

Pero sólo desde la segunda mitad del siglo XX el tema se ha venido estudiando en el ámbito escolar con el fin de determinar los momentos óptimos para el aprendizaje. Según la cronopsicología, la mente está sujeta al ciclo del sueño y la vigilia. Cuando despierta se sintoniza en un estado de vigilancia, pero no lo hace directamente, como un interruptor, sino en un proceso gradual. Por eso, ese estado varía durante el día: al principio hay somnolencia, luego mayor atención pero aún con relajación, más tarde se está atento, luego emocionado y al final de la tarde, sobreexcitado. Para

sorprende de muchos, "los mayores niveles de activación no aseguran una mayor eficiencia, afirma el experto.

Todo depende de la tarea. Las más simples, como memorizar las tablas de multiplicar o un poema, requieren de menores niveles de activación", explica Hederich. Las actividades críticas y complejas son mejores a media mañana o a la mitad de la tarde, cuando la mente aún está en un nivel intermedio, ni muy fría ni muy caliente.

Al final de la jornada, cuando el cerebro está muy excitado son más apropiadas las actividades que involucren la motricidad fina o gruesa, como practicar un deporte o una actividad artística. **Así, las actividades menos exigentes en términos intelectuales se pueden realizar en los extremos del día.**

El ritmo del cerebro es interrumpido sólo en un momento, entre las 13 y las 16 horas, después del almuerzo. "Es un descenso general en la actividad eléctrica del cerebro", explica Hederich, y se presenta aún cuando no ha habido ingesta de comida. Basados en esa teoría se puede afirmar que la siesta es un hábito sano que las sociedades modernas han ido perdiendo.

[18]

CAPITULO III.

MÉTODO HEURISTICO TABU SEARCH

1.10INTRODUCCION

En el presente capítulo describiremos el modelo de tabu search aplicado a la generación de horarios (el problema de la generación de horarios), después veremos los requerimientos y limitaciones que se necesitan para poder obtener un resultado óptimo (requerimientos y limitaciones) y por último se da la formulación matemática descrita en pseudocódigo del programa que se realizó (formulación matemática).

1.10.1 EL PROBLEMA DE LA GENERACION DE HORARIOS

Ahora describiremos el método de tabu search para el problema específico de la programación de horarios, para lo cuál definiremos nuestro problema de la siguiente manera:

Sea X_{ijk} una variable entera la cual toma el valor uno si se programa al maestro i en el horario j y se le asigna la materia k y toma el valor cero en cualquier otro caso.

Definiremos S como el conjunto de materias que se le programaron al

maestro i en los diferentes horarios que tenga disponibles, entonces tendríamos que:

$$S \leq \sum X_{irk} \text{ para } j=1,2,\dots n.$$

Definimos $C(S(x))$ como la función que genera la información de a qué maestros se le asigna qué materia y cuantas veces, por lo que:

Mínima carga académica del maestro $\square C(S(x)) \square$ Máxima carga académica del maestro.

Dado que la variable X_{ijk} toma el valor uno si se programa al maestro i en el horario j y se le asigna la materia k , entonces tendríamos que:

$$\sum X_{rjk} = 1 \text{ ó } \sum X_{rjk} = 0$$

Y dado que S representa el número de cursos programados para el maestro i , en el horario j , tendríamos:

$$S \leq \sum X_{irj} \text{ para } r=0, 1,2,\dots \text{ máxima carga académica.}$$

1.10.2 Función objetivo

El cálculo de la función objetivo esta formado por varias componentes, las cuales se desean minimizar, estas son:

- El componente académico: se tiene un costo C_{ik} de asignar al maestro

i en la materia k , en este caso mientras más deseable sea asignar a un maestro una materia, menor debe ser C_{ik} .

- La componente de horarios: una vez establecida una programación se puede medir lo compacto de ésta, midiendo el total de horas libres entre cursos para cada maestro.
- La componente de cursos distintos: se puede medir para cada maestro las materias distintas asignadas.

1.10.3 Movimientos posibles

Los movimientos posibles asignados en este problema son:

1. Subir o bajar el valor de una variable (cero o uno): Para subir el valor de una variable se requiere que la suma de variables en ese grupo sea cero y todas las variables del grupo pueden subir.
2. Subir una variable y bajar otra. Para bajar el valor de una variable se requiere que la suma del grupo sea uno y en ese grupo solo una variable pueda bajar.

Para los movimientos de las variables donde se utilizan dos variables, una que sube y otra baja se requiere que:

1. Si ambas variables están en el mismo grupo, entonces la variable que baja deberá ser la que tiene asignado el número uno en el grupo.
2. Si ambas variables están en grupos distintos se requiere entonces que la variable que baja esté localizada en el grupo cuya suma sea uno y la variable que sube puede localizarse en cualquier grupo cuya suma sea cero, esto asegura que la suma de un grupo sea cero y la suma en el otro sea uno.

El procedimiento se llevaría a cabo de la siguiente manera: Sobre todos los

grupos se revisa la suma de las variables. Si ésta vale uno se busca en las variables del grupo, cual variable vale 1 y se prueba la disminución de su valor, haciendo la variable 0. Si la suma es cero se prueba el incremento de todas las variables (una por una) en el grupo, eligiendo por último la que genera "la mejor solución". Cada uno de los movimientos realizados es almacenado en la lista $T(x)$ tabú de tal manera que el movimiento anterior no podrá realizarse otra vez a menos que cumpla con el criterio de aspiración (es decir que genera la mejor solución encontrada en el grupo).

Para los movimientos de dos variables se procede en forma similar: Se comparan las sumas de dos grupos: si sus sumas son distintas, se busca en el grupo cuya suma sea uno la variable que vale uno, para disminuir su valor a cero. Y de forma similar a la anterior, se prueba aumentando el valor de cada una de las variables del otro grupo. Si el grupo es el mismo y la suma es cero, no se hace nada. Si la suma es 1 se procede en forma similar. Cada vez que se genere un movimiento la lista tabú se ira incrementando.

La lista tabú se implementa guardando dos valores, el primero indica las variables que suben su valor y el segundo las que bajan, cuando el movimiento es de una sola variable se puede representar como $(0,1)$ si su valor baja ó $(1,0)$ si su valor sube.

1.10.4 REQUERIMIENTOS Y LIMITACIONES

Los aspectos que se consideraron en la programación son:

1. Número de cursos a programar
2. Total de salones disponibles

3. Total de cursos para asignar a cada maestro
4. Cursos preestablecidos a ciertos maestros
5. Los horarios preestablecidos a ciertas materias
6. Programación de un número determinado de materias en el horario matutino y otro número determinado en el horario vespertino,
7. La compactación de horarios

Como se puede ver todos estos aspectos generan un modelo grande respecto al número de variables y de restricciones que se manejan, sin embargo un acomodo adecuado de las estructuras de datos nos simplificó su implementación. Además el número de movimientos posibles se disminuye en gran medida por las restricciones que se establecen para los grupos que obligan a que cada maestro que no se le pueda asignar dos materias en el mismo horario es decir que $\sum X_{rjk} = 1$, es el máximo valor que puede tomar.

1.10.5 FORMULACIÓN MATEMÁTICA (pseudocódigo)

INTRODUCIR DATOS ← BASE DE DATOS INICIAL INICIA {

PASO 1. Formar los grupos de cada maestro

PASO 2. Seleccionar un X_{ijk} en el primer grupo

Si $X_{ijk}=0$ entonces sube, de otra forma baja

PASO 3. Guardar el movimiento en la lista tabú.

PASO 4. Verificar si $\sum X_{rjk} = 1$ ó $\sum X_{rjk} = 0$

Si $\sum X_{rjk} = 1$ continuar con el siguiente grupo Si $\sum X_{rjk} = 0$ regresar al PASO 2

PASO 5. Verificar si los cursos asignados al maestro i se encuentran en el rango

$$L_1(i) \leq \sum_{i,k} X_{rjk} \leq L_1(i)$$

Si se cumple, continuar con el siguiente PASO Si no, regresar al PASO 2

PASO 6. Medir la compactación de horarios del maestro i en el horario j

$$\sum_j X_{rjk} \text{ para los } j\text{'s donde } X_{rjk} \neq 0$$

Si $\sum X_{rjk}$ es mínima:

catalogar este arreglo como ÓPTIMO

hacer los demás movimientos tabú

continuar con el siguiente PASO

PASO 7. Medir el costo académico de los ÓPTIMOS del paso anterior

$$\sum C_{ijk} X_{rjk} \text{ i,j,k}$$

Hacer ÓPTIMO el arreglo que tenga el menor costo.

PASO 8. Verificar si el número de salones disponibles en el horario j anda en el rango

$$L_2(j) \leq \sum_{j,k} X_{rjk} \leq L_2(j)$$

Si no se cumple regresar al PASO 2

Si se cumple continuar con el siguiente PASO

PASO 9. Verificar si en el horario j se programaron $C_2(j)$ cursos de la materia k

$$\sum_i X_{rjk} \geq C_2(j)$$

Si no se cumple regresar al PASO 2

Si se cumple continuar con el siguiente PASO

PASO 10. Verificar si para la materia k se programaron los cursos $C_3(k)$ en la mañana y $C_4(k)$ en la tarde, tomando en cuenta que L es el último horario de la mañana.

$$\sum_i X_{rjk} = C_3(j) \quad \text{con } j \leq L$$

$$\sum_i X_{rjk} = C_4(j) \quad \text{con } j \geq L$$

TERMINA.

Si no se cumple regresar al PASO 2.

Si se cumple catalogar este arreglo como la MEJOR SOLUCIÓN

...[17]

1.11 APLICACIÓN DEL MÉTODO HEURÍSTICO TABU SEARCH EN LA GENERACION DE HORARIOS

Para poder Desarrollar el presente trabajo se formulo un algoritmo de Tabu Search para poder realizar la generación de Horarios de manera eficiente, para lo cual se definieron:

1.11.1 Entidades:

Memoria Tabu (mT)

Es el número de pasos que la Matriz Tabú recordá para que no se vuelva a pasar en los últimos mT Movimientos.

Tiempos o Periodos (T)

El conjunto de Tiempos para el Horario de estudio lo consideraremos como un horario escalar continuo es decir si un horario es de Lunes a Viernes (5 días) y cada día hay 7 periodos entonces todo el horario sería considerado como horas del 1 al 35 ($nt = \text{días} \times \text{periodos}$). Y lo representaremos como un vector T que tiene nt elementos

$$T = \{t_1, t_2, t_3, \dots, t_{nt}\}$$

En forma General cada Elemento de T lo representamos mediante t_i .

Profesores (P)

El conjunto de Profesores lo representamos como un Vector P que tiene np elementos. Cada Elemento del vector P lo representaremos como p_j .

$$P = \{p_1, p_2, p_3, \dots, p_{np}\}$$

En forma General cada Elemento de P lo representamos mediante p_j .

Secciones (S)

El conjunto de Secciones que hay en todo el Colegio como por ejemplo Secundaria 5to A, Primaria 5to B, etc. Y lo representamos mediante un vector S que tiene ns elementos. Cada elemento de S lo representamos mediante s_k .

$$S = \{s_1, s_2, s_3, \dots, s_{ns}\}$$

En forma General cada Elemento de S lo representamos mediante s_k .

Cursos (C)

El conjunto de Cursos o Asignaturas que se emplean en un horario como por ejemplo Matemática, Literatura, Tutoría, etc. Y lo representamos mediante un Vector C que tiene nc elementos. Cada elemento de C lo representamos mediante c_j .

$$C = \{c_1, c_2, c_3, \dots, c_{nc}\}$$

En forma General cada Elemento de C lo representamos mediante c_j .

Matrices Necesarias:

Relaciones Profesor-Curso-Sección

Esta Matriz Define las Asignaciones Iniciales de Cada Profesor con el Curso que imparte en su respectiva sección indica además el numero de horas que impartirá en ese determinado Curso-Sección el numero de relaciones será **nr.**

$$RPCS = \begin{matrix} & \begin{matrix} Profesor & Curso & Seccion & nHoras \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \dots \\ nr \end{matrix} & \begin{bmatrix} p4 & c6 & s14 & 2 \\ p4 & c5 & s15 & 7 \\ \dots & \dots & \dots & \dots \\ p53 & c7 & s19 & 2 \end{bmatrix} \end{matrix}$$

Esta matriz sirve para alimentar con la información necesaria a las matrices Profesor-Sección y Profesor-Curso donde se ejecutaran los algoritmos.

Asignaciones Profesor – Sección

Usaremos la matriz PS para poder tener la información de cada Profesor en que Sección está asignado a lo largo de todos los periodos T.

$$PS = \begin{matrix} & \begin{matrix} t1 & t2 & \dots & t_{nt} \end{matrix} \\ \begin{matrix} p1 \\ p2 \\ \dots \\ p_{np} \end{matrix} & \begin{bmatrix} 0 & s1 & * & s3 \\ s1 & 0 & 0 & 0 \\ 0 & 0 & 0 & s1 \\ s2 & 0 & s3 & 0 \end{bmatrix} \end{matrix}$$

Donde cada Fila representa el horario de cada profesor, por ejemplo para el profesor p1 tiene asignado la sección s1 en la segunda hora y s3 en la última. El "0" de la primera hora indica que el profesor esta libre a esa hora y el "*" de la tercera nos indica que el profesor está restringido es decir que no puede dictar a esa hora, en términos prácticos se cambiará el "*" que indica el bloqueo por un valor -1, ya que la matriz será una matriz de números enteros:

Asignaciones Profesor – Curso

Usaremos la matriz PC para poder tener la información de cada Profesor en que Curso está asignado a lo largo de todos los periodos T.

$$PC = \begin{matrix} & t1 & t2 & \dots & t_{nt} \\ p1 & \begin{bmatrix} 0 & c2 & * & c1 \end{bmatrix} \\ p2 & \\ \dots & \\ p_{np} & \end{matrix}$$

Donde cada Fila representa el horario de cada profesor, por ejemplo para el profesor p1 tiene asignado el sección c2 en la segunda hora y c1 en la última. El "0" de la primera hora indica que el profesor esta libre a esa hora y el "*" de la tercera nos indica que el profesor está restringido es decir que no puede dictar a esa hora.

Esta matriz es complementaria a la Matriz Profesor–Sección, ya que sirve para recordar que curso estaba asignado en la sección ya definida. Por ejemplo podemos afirmar que el profesor p1 enseña en la segunda hora en la sección s1 (Matriz PS) el curso c2 (Matriz PC), y en la ultima hora enseña en la sección s3 el curso c1.

Memoria Profesor-Tabú

Esta Matriz es usada para recordar los últimos n movimientos a fin de que no se vuelva a recorrer por los últimos mT (Memora Tabú) caminos ya trazados.

$$PT = \begin{matrix} & t1 & t2 & \dots & t_{nt} \\ p1 & \begin{bmatrix} 0 & 1 & * & 1 \end{bmatrix} \\ p2 & \\ \dots & \\ p_{np} & \end{matrix}$$

Esta matriz evita que la búsqueda de la mejor solución quede en una vecindad y no busque otras posibles soluciones.

Bloqueo de Secciones (chkSec)

Esta Matriz es usada para almacenar los cruces de profesores por cada aula a lo largo de todos los periodos T.

$$chkSec = \begin{matrix} & & t1 & t2 & \dots & t_{nt} \\ s1 & \left[\begin{array}{cccc} 2 & 1 & 0 & 1 \end{array} \right. \\ s2 & \left. \begin{array}{cccc} 1 & 0 & 1 & 0 \end{array} \right. \\ \dots & \left. \begin{array}{cccc} \dots & \dots & \dots & \dots \end{array} \right. \\ s_{ns} & \left. \begin{array}{cccc} 2 & 0 & 2 & 0 \end{array} \right. \end{matrix}$$

Esta matriz mantiene actualizados las posiciones donde existen cruces entre profesores (bloqueos), por ejm en $chkSec[0,0] = 2$ existe un cruce entre 2 profesores y en $chkSec[0,2] = 0$ No hay asignado ningún profesor en esa aula en ese periodo de clases.

Función Objetivo

Para poder tratar la Distribución Horaria como un problema de optimización, es necesario definir una función objetivo que determina el grado de inviabilidad y la insatisfacción de las necesidades, es decir, pretende generar soluciones factibles con un número mínimo de requisitos insatisfechas. Así, cada Horario H se evaluó con la siguiente función objetivo, que debe ser minimizado:

$$F(H) = k1 \times F1(H) + k2 \times F2(H) + k3 \times F3(H) + k4 \times F4(H) + k5 \times F5(H),$$

donde :

F1, para cada periodo, el número de veces que más de un profesor enseña en la misma clase en el por cada periodo k y el número de veces que una clase no tiene actividad.

F2, mide el número de asignaciones que no respetan los límites diarios de lecciones.

Como tal, una distribución horaria calendario sólo puede considerarse factible si $F1(H) = F2(H) = 0$.

F3 mide el número de cursos de ciencias que son fijados en la banda inferior (de menor rendimiento académico).

F4 mide el número de cursos de ciencias que son fijados en la banda secundaria (de rendimiento medio).

F5 mide el nivel de dispersión horaria, los cuales son fijados como el cociente entre la sección de mayor dispersión con la de menor dispersión.

1.11.2 ALGORITMO DEL MÉTODO DE TABU SEARCH PARA LA PROGRAMACIÓN DE HORARIOS (PSEUDOCÓDIGO)

INICIO

CreaMatriz();

/* Se crea las Variable Globales y Matrices Necesarias para poder ejecutar el algoritmo */

InicializaMatriz();

/* Se Procede a la carga e Inicialización de las matrices Principales PS y PC asi como MT */

numIntentos ← 2000:

numCiclos ← 1000:

numCiclosAleatorios ← 80:

numCiclosQuitaBloqueos ← 40:

Proceso(numIntentos, numCiclos , numCiclosAleatorios,
numCiclosQuitaBloqueos);

/* Primera Ejecución en afinamiento Masivo */

numIntentos ← 2000:

numCiclos ← 1000:

numCiclosAleatorios ← 40:

numCiclosQuitaBloqueos ← 20:

Proceso(numIntentos, numCiclos , numCiclosAleatorios,
numCiclosQuitaBloqueos);

/* Segunda Ejecución en afinamiento Medio */

numIntentos ← 2000:

numCiclos ← 1000:

numCiclosAleatorios ← 20:

numCiclosQuitaBloqueos ← 10:

Proceso(numIntentos, numCiclos , numCiclosAleatorios,
numCiclosQuitaBloqueos);

/* Tercera Ejecución en afinamiento de corto alcance */

MuestraPSFinal();

/* Muestra de resultados de las matrices obtenidas así como también de las funciones de costos intermedias obtenidas durante el proceso */

FIN

SUBVOID Proceso(numIntentos, numCiclos, numCiclosAleatorios,
numCiclosQuitaBloqueos)

INICIO

CalculaCosto();

/* Calculo de Costo Inicial */

CargaPS("I");

/* Carga de matriz Final y Función Costos por primera vez */

FOR iIntento = 0, numIntentos -1;

/* Iteraciones de Intentos Completos */

INICIO

LimpiaTabu();

/* Borra la memoria Tabu */

FOR iCiclo = 0, numCiclos -1;

/* Iteraciones de Intentos de Intentos Completos antes de borrar Memoria Tabú */

INICIO

CalculaCosto();

/* Calculo de Costo de la Matriz en el momento actual */

FOR iCQB = 0,

ALEATORIO(0,numCiclosQuitaBloqueos -1);

/* Ciclo de Búsquedas a través del núcleo Bloqueos */

INICIO

BuscaAleatorio(ref Prnd, ref T1rnd, ref
T2rnd);

BuscaBloqueo(ref Prnd, ref T1rnd, ref T2rnd);

Intercambia(Prnd, T1rnd, Prnd, T2rnd);

CalculaBloqueo();

IF (numBloqueo = 0)

CONTINUAR ;

FIN

```
CalculaCosto();
```

```
IF (numCosto <= numCostoFinal)
```

```
/* Si el Costo es menor entonces se Actualiza la matriz Final y el numCostoFinal */
```

```
    CargaPS("B");
```

```
ELSE
```

```
    RecuperaPS();
```

```
FIN
```

```
FOR iCAL = 0, ALEATORIO(0, numCiclosAleatorios-1);
```

```
/* Ciclo de Búsquedas a través del núcleo Aleatorios */
```

```
    INICIO
```

```
        BuscaAleatorio(ref Prnd, ref T1rnd, ref  
T2rnd);
```

```
        Intercambia(Prnd, T1rnd, Prnd, T2rnd);
```

```
        CalculaBloqueo();
```

```
        IF (numBloqueo = 0)  
            CONTINUAR ;
```

```
        FIN
```

```
    CalculaCosto();
```

```
IF (numCosto <= numCostoFinal)
```

```
/* Si el Costo es menor entonces se Actualiza la matriz Final y el numCostoFinal */
```

```
    CargaPS("B");
```

```
ELSE
```

```
    RecuperaPS();
```

```
FIN
```

```
FIN
```

```
FIN
```

SUBVOID BuscaAleatorio(ref int pi, ref int t1, ref int t2)

/* Genera 2 coordenadas que no estén en Tabu ,de un profesor para que estos intercambien sus asignaciones*/

SUBVOID BuscaBloqueo(ref int pi, ref int t1, ref int t2)

/* Busca el Bloqueo mas cercano a partir de un aposicion en a matriz chkSec */

SUBVOID Intercambia(int p1, int t1, int p2, int t2)

/* Intercambia 2 asignaciones de un profesor y manda a grabar en la memoria matu ese par de asignaciones*/

SUBVOID CargaTabu(int p1, int t1, int p2, int t2)

/* Carga en la Memoria Tabu 2 asignaciones dadas */

SUBVOID LimpiaTabu()

/* Vuelve a 0 la Memoria Tabú */

SUB LOGICO EsTabu(int p1, int t1)

/* Verifica es elemento d ela matriz ya esta en la memora tabu */

SUBVOID CreaMatriz()

/* AQUI Se crean las Matrices Principales y se cargan lo datos Iniciales*/

SUBVOID CargaPS(String cadTipoAsig)

/* Carga de la Matriz Temporal PS a PSFinal si como su num bloques*/

SUBVOID RecuperaPS()

/* Recupera de PSFinal a PS para volver a procesar*/

SUBVOID CalculaBloqueo()

/* Calcula la matriz de bloqueos chkSec y calcula numBloqueos */

SUBVOID CalculaCosto()

/* Calcula el costos de PS y lo almacena en la variable publica numCosto */

SUBVOID AsignaHora(int Prof, int Curso, int Seccion)

/* Carga por primera vez desde la matriz de Entrada RPCS hacia la matriz PS y PC dándole cierto grado de aleatoriedad */

CAPITULO IV

ANALISIS DE LOS RESULTADOS

1.12 PRESENTACION E INTERPRETACION DE RESULTADOS

Se cargo al Sistema los Siguietes datos de Entrada:

Listado de Profesores

```
// PROFESOR
#region LLenado de Profesor
Profesor[000] = "ROMÁN SÁENZ, CLARA ALICIA";
Profesor[001] = "CARLOS AMESQUITA, LEONARDO ";
Profesor[002] = "ZAMORA RAMOS, NORMAN REGIS";
Profesor[003] = "PEÑALOZA FLORES, KAREN MISHHELL";
Profesor[004] = "CAMARA PALOMINO, KEVIN ";
Profesor[005] = "RETAMOZO SANCHEZ, GRISELDA ";
Profesor[006] = "PAZ CCORICASA, NELLY";
Profesor[007] = "PAULINO FLORES, LUIS ALBERTO";
Profesor[008] = "CANALES BIELICH, MERCHE XIMENA";
Profesor[009] = "VILLANUEVA TERRONES, ARTURO ";
Profesor[010] = "RAMIREZ BUIZA, CARLOS LEONEL";
Profesor[011] = "VILCA PIZARRO, GLORIA MARCELINA";
Profesor[012] = "CARRANZA CHIARA, ANA LUCIA";
Profesor[013] = "AMAYA CAMAYOC, SAMUEL JESUS";
Profesor[014] = "PALACIOS ORÓS, JUAN JOSÉ";
Profesor[015] = "ORTÍZ FARRO, ALMENDRA RUBI";
Profesor[016] = "QUISPE AGUIRRE, LUISA STEPHANY";
Profesor[017] = "CAJUSOL BALDEON, GINA DEL ROCIO";
Profesor[018] = "AGUIRRE CALDERON, SHIRLEY TANIA";
Profesor[019] = "PANDO GARCIA, RENATO LEONCIO";
Profesor[020] = "PALACIOS CANORIO, IRINA ADRIANA";
Profesor[021] = "ROSADA CAHUANA, ROXANA";
Profesor[022] = "VÁSQUEZ DÍAZ, THALIA";
Profesor[023] = "ALIAGA CIPRIAN, VICTOR LEONARDO";
Profesor[024] = "PIZARRO PARHUAYO, MELISSA JESSICA";
Profesor[025] = "AGUILAR GALLEGOS, MARYORY JHOSELIN";
Profesor[026] = "VIZCARDO HILARES, AYLEEM LUISA";
Profesor[027] = "ZEGARRA SUSONI, LÓUIS ALBERTH";
Profesor[028] = "CARDENAS ARRIETA, JEAN FRANCO";
```

```

Profesor[029] = "CRISPIN PAZ, BRIYIT MIDORY";
Profesor[030] = "ROMERO ESPINOZA, KAREN";
Profesor[031] = "PALMA TAPIA, SUSY IRENE";
Profesor[032] = "SOLORZANO PONCE, AMELIA CONSTANTINA";
Profesor[033] = "ALPES LEANDRO, ANIVAR";
Profesor[034] = "ROJAS RIMARI, JOHAN MAX";
Profesor[035] = "BORJA HUAMANI, ANGÉLICA ANELI";
Profesor[036] = "VIERA NINANYA, LISETH MARCELA";
Profesor[037] = "RAMOS USQUIANO, ANDREA ARACELLI";
Profesor[038] = "CABALLERO OSORIO, GERSON REMIGIO";
Profesor[039] = "QUISPE CALDERÓN, DEBORAH LISSET";
Profesor[040] = "PAZ RIVAS, FERNANDO GUSTAVO";
Profesor[041] = "CAMPOS LÓPEZ, JULIO CÉSAR";
Profesor[042] = "CERRÓN MARQUINA, ELSA BEATRIZ";
Profesor[043] = "CARLOS INGARUCA, EILEEN GEMMA";
Profesor[044] = "RAMOS SUCA, CAROLINA LUZ";
Profesor[045] = "BUSTOS CASTILLO, ALFREDO ALEXANDER";
Profesor[046] = "CHAVEZ CABALLERO, DIANE ALEXANDRA";
Profesor[047] = "POMA NAVARRO, LUIS JAVIER";
Profesor[048] = "BECERRA ALARCÓN, EDUARDO DAVID";
Profesor[049] = "ROJAS SANTIBAÑEZ, SHAROW RUTH";
Profesor[050] = "SALAS LEON, Orlando Rolando";
Profesor[051] = "PALOMINO VILLEGAS, JONATHAN JOSE";
Profesor[052] = "CONDORI MAMANI, CESAR ELIO";
#endregion

```

Listado de Cursos

```

// CURSO
#region LLenado de Curso
Curso[000] = "Comunicación";
Curso[001] = "Mat";
Curso[002] = "CTA";
Curso[003] = "HGyE";
Curso[004] = "PerFamyRH";
Curso[005] = "FormCiudyCiv";
Curso[006] = "Tutoria";
Curso[007] = "EdTrabajo";
Curso[008] = "Arte";
Curso[009] = "E.Fisica";
Curso[010] = "Edregiliosa";
Curso[011] = "Ingles";
#endregion

```

Listado de Secciones

```

//SECCION
#region LLenado de Seccion
Seccion[000] = "S1A";
Seccion[001] = "S1B";
Seccion[002] = "S1C";
Seccion[003] = "S1D";
Seccion[004] = "S1E";
Seccion[005] = "S1F";
Seccion[006] = "S1G";
Seccion[007] = "S1H";
Seccion[008] = "S3A";
Seccion[009] = "S3B";

```

```
Seccion[010] = "S3C";  
Seccion[011] = "S3D";  
Seccion[012] = "S3E";  
Seccion[013] = "S3F";  
Seccion[014] = "S4A";  
Seccion[015] = "S4B";  
Seccion[016] = "S4C";  
Seccion[017] = "S4D";  
Seccion[018] = "S4E";  
Seccion[019] = "S4F";  
Seccion[020] = "S5A";  
Seccion[021] = "S5B";  
Seccion[022] = "S5C";  
Seccion[023] = "S5D";  
Seccion[024] = "S5E";  
Seccion[025] = "S5F";  
#endregion
```

Relacion Profesor-Curso-Seccion-Nro de Horas

Por ejemplo en la Primera Fila significa que el profesor de código 000 (ROMÁN SÁENZ, CLARA ALICIA) enseña el curso de código 010 (Edregiliosa), en la Sección 020 (55A) un total de 2 horas.

ProfCurSec [000,0]=000;	ProfCurSec [000,1]=010;	ProfCurSec [000,2]=020;	ProfCurSec [000,3]=002;
ProfCurSec [001,0]=000;	ProfCurSec [001,1]=010;	ProfCurSec [001,2]=021;	ProfCurSec [001,3]=002;
ProfCurSec [002,0]=000;	ProfCurSec [002,1]=006;	ProfCurSec [002,2]=023;	ProfCurSec [002,3]=001;
ProfCurSec [003,0]=001;	ProfCurSec [003,1]=001;	ProfCurSec [003,2]=024;	ProfCurSec [003,3]=006;
ProfCurSec [004,0]=001;	ProfCurSec [004,1]=001;	ProfCurSec [004,2]=025;	ProfCurSec [004,3]=006;
ProfCurSec [005,0]=002;	ProfCurSec [005,1]=000;	ProfCurSec [005,2]=020;	ProfCurSec [005,3]=005;
ProfCurSec [006,0]=002;	ProfCurSec [006,1]=000;	ProfCurSec [006,2]=014;	ProfCurSec [006,3]=007;
ProfCurSec [007,0]=003;	ProfCurSec [007,1]=005;	ProfCurSec [007,2]=005;	ProfCurSec [007,3]=002;
ProfCurSec [008,0]=003;	ProfCurSec [008,1]=005;	ProfCurSec [008,2]=007;	ProfCurSec [008,3]=002;
ProfCurSec [009,0]=003;	ProfCurSec [009,1]=004;	ProfCurSec [009,2]=002;	ProfCurSec [009,3]=002;
ProfCurSec [010,0]=003;	ProfCurSec [010,1]=004;	ProfCurSec [010,2]=006;	ProfCurSec [010,3]=002;
ProfCurSec [011,0]=004;	ProfCurSec [011,1]=002;	ProfCurSec [011,2]=020;	ProfCurSec [011,3]=006;
ProfCurSec [012,0]=004;	ProfCurSec [012,1]=002;	ProfCurSec [012,2]=021;	ProfCurSec [012,3]=006;
ProfCurSec [013,0]=005;	ProfCurSec [013,1]=002;	ProfCurSec [013,2]=004;	ProfCurSec [013,3]=006;
ProfCurSec [014,0]=005;	ProfCurSec [014,1]=002;	ProfCurSec [014,2]=005;	ProfCurSec [014,3]=006;
ProfCurSec [015,0]=006;	ProfCurSec [015,1]=000;	ProfCurSec [015,2]=004;	ProfCurSec [015,3]=005;
ProfCurSec [016,0]=006;	ProfCurSec [016,1]=000;	ProfCurSec [016,2]=005;	ProfCurSec [016,3]=005;
ProfCurSec [017,0]=007;	ProfCurSec [017,1]=000;	ProfCurSec [017,2]=023;	ProfCurSec [017,3]=005;
ProfCurSec [018,0]=007;	ProfCurSec [018,1]=000;	ProfCurSec [018,2]=024;	ProfCurSec [018,3]=005;
ProfCurSec [019,0]=007;	ProfCurSec [019,1]=000;	ProfCurSec [019,2]=015;	ProfCurSec [019,3]=007;
ProfCurSec [020,0]=007;	ProfCurSec [020,1]=000;	ProfCurSec [020,2]=019;	ProfCurSec [020,3]=007;
ProfCurSec [021,0]=008;	ProfCurSec [021,1]=000;	ProfCurSec [021,2]=002;	ProfCurSec [021,3]=005;
ProfCurSec [022,0]=008;	ProfCurSec [022,1]=000;	ProfCurSec [022,2]=003;	ProfCurSec [022,3]=005;
ProfCurSec [023,0]=009;	ProfCurSec [023,1]=000;	ProfCurSec [023,2]=010;	ProfCurSec [023,3]=007;
ProfCurSec [024,0]=009;	ProfCurSec [024,1]=000;	ProfCurSec [024,2]=011;	ProfCurSec [024,3]=007;
ProfCurSec [025,0]=009;	ProfCurSec [025,1]=000;	ProfCurSec [025,2]=012;	ProfCurSec [025,3]=007;
ProfCurSec [026,0]=009;	ProfCurSec [026,1]=000;	ProfCurSec [026,2]=025;	ProfCurSec [026,3]=005;
ProfCurSec [027,0]=010;	ProfCurSec [027,1]=000;	ProfCurSec [027,2]=000;	ProfCurSec [027,3]=005;
ProfCurSec [028,0]=010;	ProfCurSec [028,1]=000;	ProfCurSec [028,2]=001;	ProfCurSec [028,3]=005;
ProfCurSec [029,0]=010;	ProfCurSec [029,1]=000;	ProfCurSec [029,2]=008;	ProfCurSec [029,3]=007;
ProfCurSec [030,0]=010;	ProfCurSec [030,1]=000;	ProfCurSec [030,2]=009;	ProfCurSec [030,3]=007;
ProfCurSec [031,0]=010;	ProfCurSec [031,1]=006;	ProfCurSec [031,2]=022;	ProfCurSec [031,3]=001;
ProfCurSec [032,0]=011;	ProfCurSec [032,1]=000;	ProfCurSec [032,2]=006;	ProfCurSec [032,3]=005;
ProfCurSec [033,0]=011;	ProfCurSec [033,1]=000;	ProfCurSec [033,2]=007;	ProfCurSec [033,3]=005;
ProfCurSec [034,0]=012;	ProfCurSec [034,1]=000;	ProfCurSec [034,2]=017;	ProfCurSec [034,3]=007;
ProfCurSec [035,0]=012;	ProfCurSec [035,1]=000;	ProfCurSec [035,2]=018;	ProfCurSec [035,3]=007;
ProfCurSec [036,0]=012;	ProfCurSec [036,1]=000;	ProfCurSec [036,2]=021;	ProfCurSec [036,3]=005;
ProfCurSec [037,0]=012;	ProfCurSec [037,1]=000;	ProfCurSec [037,2]=022;	ProfCurSec [037,3]=005;
ProfCurSec [038,0]=013;	ProfCurSec [038,1]=000;	ProfCurSec [038,2]=013;	ProfCurSec [038,3]=007;
ProfCurSec [039,0]=013;	ProfCurSec [039,1]=000;	ProfCurSec [039,2]=016;	ProfCurSec [039,3]=007;
ProfCurSec [040,0]=013;	ProfCurSec [040,1]=006;	ProfCurSec [040,2]=013;	ProfCurSec [040,3]=001;
ProfCurSec [041,0]=014;	ProfCurSec [041,1]=002;	ProfCurSec [041,2]=022;	ProfCurSec [041,3]=006;
ProfCurSec [042,0]=014;	ProfCurSec [042,1]=002;	ProfCurSec [042,2]=023;	ProfCurSec [042,3]=006;
ProfCurSec [043,0]=014;	ProfCurSec [043,1]=002;	ProfCurSec [043,2]=024;	ProfCurSec [043,3]=006;
ProfCurSec [044,0]=014;	ProfCurSec [044,1]=002;	ProfCurSec [044,2]=025;	ProfCurSec [044,3]=006;
ProfCurSec [045,0]=014;	ProfCurSec [045,1]=006;	ProfCurSec [045,2]=025;	ProfCurSec [045,3]=001;
ProfCurSec [046,0]=016;	ProfCurSec [046,1]=001;	ProfCurSec [046,2]=000;	ProfCurSec [046,3]=006;
ProfCurSec [047,0]=016;	ProfCurSec [047,1]=001;	ProfCurSec [047,2]=001;	ProfCurSec [047,3]=006;
ProfCurSec [048,0]=016;	ProfCurSec [048,1]=001;	ProfCurSec [048,2]=002;	ProfCurSec [048,3]=006;
ProfCurSec [049,0]=016;	ProfCurSec [049,1]=001;	ProfCurSec [049,2]=003;	ProfCurSec [049,3]=006;
ProfCurSec [050,0]=017;	ProfCurSec [050,1]=001;	ProfCurSec [050,2]=010;	ProfCurSec [050,3]=006;
ProfCurSec [051,0]=017;	ProfCurSec [051,1]=001;	ProfCurSec [051,2]=011;	ProfCurSec [051,3]=006;
ProfCurSec [052,0]=017;	ProfCurSec [052,1]=001;	ProfCurSec [052,2]=012;	ProfCurSec [052,3]=006;
ProfCurSec [053,0]=017;	ProfCurSec [053,1]=001;	ProfCurSec [053,2]=013;	ProfCurSec [053,3]=006;
ProfCurSec [054,0]=018;	ProfCurSec [054,1]=001;	ProfCurSec [054,2]=016;	ProfCurSec [054,3]=006;
ProfCurSec [055,0]=018;	ProfCurSec [055,1]=001;	ProfCurSec [055,2]=017;	ProfCurSec [055,3]=006;
ProfCurSec [056,0]=018;	ProfCurSec [056,1]=001;	ProfCurSec [056,2]=018;	ProfCurSec [056,3]=006;
ProfCurSec [057,0]=018;	ProfCurSec [057,1]=001;	ProfCurSec [057,2]=019;	ProfCurSec [057,3]=006;
ProfCurSec [058,0]=019;	ProfCurSec [058,1]=001;	ProfCurSec [058,2]=008;	ProfCurSec [058,3]=006;
ProfCurSec [059,0]=019;	ProfCurSec [059,1]=001;	ProfCurSec [059,2]=009;	ProfCurSec [059,3]=006;
ProfCurSec [060,0]=019;	ProfCurSec [060,1]=001;	ProfCurSec [060,2]=014;	ProfCurSec [060,3]=006;

Después de haber realizado las pruebas necesarias se llegó a los siguientes resultados:

Periodos t = 034

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
Prof00										20											20		23				21							21		
Prof01				24					25		24	25					25	25	24	25					24							24	25			24
Prof02				14				14						20	14		20		14			20	14	14	14			20					20			
Prof03												7							2		6	2	5					6							7	5
Prof04				20		20					21	20				20		20		21	21			21		20								21	21	
Prof05		5	4	4	4				5							5	4	5	5				4	5								4				
Prof06	4		5	5	5		4				4	5	5					4										4								
Prof07	19		24	19		15	24		15	15	15	23	23		24	15			23		19	19		19	19	15	24	23		19	15		23	24		
Prof08				2				2	3	3					2						3			2			2		3					3		
Prof09	12	10	25	10	12	10	11	12	10	25	25	12				11	12	10	11	11	10	11	11	11	12	25				25	12				10	
Prof10	9	9			0	0	1	1	8		1	8	9	9		8		8	1	9	0	0	9		8			9	22			1	8	0	8	
Prof11					6		6					7	6	7									7	7			7	6							6	
Prof12	17	17			22	17	18		21	22	17	21	18	18	21	21		17		22	18	18	17	18		22	22			21	18			17		
Prof13	16			16			16		16			13	16		13				13				13				16			13	13	13		13	16	
Prof14	25	22	23	25	25	22			22	24	22	24		24	25	25	24		25	23	24	23	22	23	25			24			23	23			22	
Prof15																																				
Prof16		1	3	1	3	1	2		1		2		2	1		3	3	0	0			2	2		1	0			0	3	2	0	0		3	
Prof17	13	12	12	13	11	11	10		13	13	11	11	12	11		12	13						12	10	10		10	13		10	10	11		12		
Prof18		19	16	17	19	16			17	18	19	19	17		16		18	16	18	17			18	18	19		16	19	17	17	16	18				
Prof19	8	8	15	8	15		14	8	9		9			14	9		14	9	8		14	14			15	8			15	14	14	15	15	9	9	
Prof20	7	6		6	7	5		5	4	7	5	4	6		4	7	6		7		5	6			6	5		5		7		4	4	4		
Prof21																																				
Prof22		14	18	18	16	14			18	19		14		19	17	18	17	19	15				17	19	15	16	16		15	17	15	16	14			
Prof23		3		3	2			0	0	1	0	1	1	2	3	0	1	1		2			3	3	1	2	3			2	0		2		0	
Prof24																																				
Prof25		7	13	7	13		7	9	12	8	8	9		13		13	8							12	7	9	9	12	9	8	7	12	12		7	
Prof26		23	11	22		24			24			10			23	11		15	11	10		25		10	10		11	11						22	25	
Prof27	20	25	21	23				21		23	16	17		25	20	16	21		22	16			24		22	24	20		24	23		17	25	22	17	
Prof28	15	11		15	10	9	12	11	14		14	15	13		12				9		8	8		8		13		13	12	10		11	9	10	14	
Prof29	22	24		21	18	19	17	15		14	23		25		18		19	21	16	14	23	25	20		20		15		16	24	22		17			
Prof30	3	0				7	6		6	0	3		0			6	2	3	4		1	1	1		5	2	4	7	4	5	5		7		2	
Prof31	1		0										3	7	1	1						4			3			5		5		0	7		4	
Prof32							18			18				19	19													19							18	
Prof33	10		9	11						12	13				8								9	10			11		8		12				13	
Prof34	18	16		24		21		20		17			19	22	23	17		22		15	15	16	21	20		14	23	25	18		25	19	24	14		
Prof35	0	2	6	0	1		9	10		4			8	8	0		9		3	13	6				4	12	13	2	11	1	11	3	10	12		
Prof36	23					4	22	25	7	6		2	22	0	5	24	5	6		4	7		0	24	23		3	3	25				2	1	1	
Prof37			10		9	8				10												8				9										
Prof38			14	12	17		21	13	20	11			20	16		14	16	15		12	11	15		16		17	19	17		18		21	13	19	18	
Prof39	14	13			14		8	19				10	15	17	15	10	11	13	12	18	12		16		9		17	11		16	9	18		8	19	
Prof40						2	3	4			7			4			0					3	5			1	1	0	7	2	6	5	6			
Prof41	24		22			20	25								21				23		20		24			21	23	25	22							

Prof42	11		19		13	13	17	11	9		16	14	12	10		18		10	17	12	8				18	14	8	9	19		16	15	15		
Prof43	2	4		9	8	12	0				0	10	3		4	10	12	2	1		9	13	4	13		8		1			8	3	11	11	
Prof44	21	15	17		23	19	22		16			24	15	22			20	24	16	21		17			18	14	18	14	20	21	24	19	23		
Prof45		21			21												17					25	25	17				20				20	20		
Prof46			2		3	5	7	24	2		3		5	6	22	23	24	6	7	22	4		0	0	6			23	4	1		1	5		
Prof47		18	8		18	15	16	19		12	18	11	10		9		14	19	19	13		15	13			12		10	11	8	9	14	16		
Prof48	6		1		6		3	2	5	6	6	4			2		7		0		7	6	6	3	4	0	1	6		3	6	5	2		
Prof49		20	20	20	23		23	23	23	21	20	22	21		23	22		21			22		22		21		21	21	22	20	22	20		23	
Prof50																																			
Prof51	5		7		25											7		5							7				6				6	25	
Prof52																										10									

MATRIZ PS - HORARIO DE CADA PROFESOR EN QUE SECCION LE CORRESPONDE ENSEÑAR

Donde el valor en cada casilla corresponde a la sección de debe enseñar.

Así mismo tenemos la Matriz PC que corresponde a que curso debe enseñar cada profesor en el horario calculado, en este caso los cursos 1 y 2 corresponden a los cursos de Matemática y CTA respectivamente. Se puede apreciar que la mayor concentración horaria de estos ahora se encuentra en la franja horaria de mayor posibilidad de aprendizaje. (3ra, 4ta y Quinta Hora).

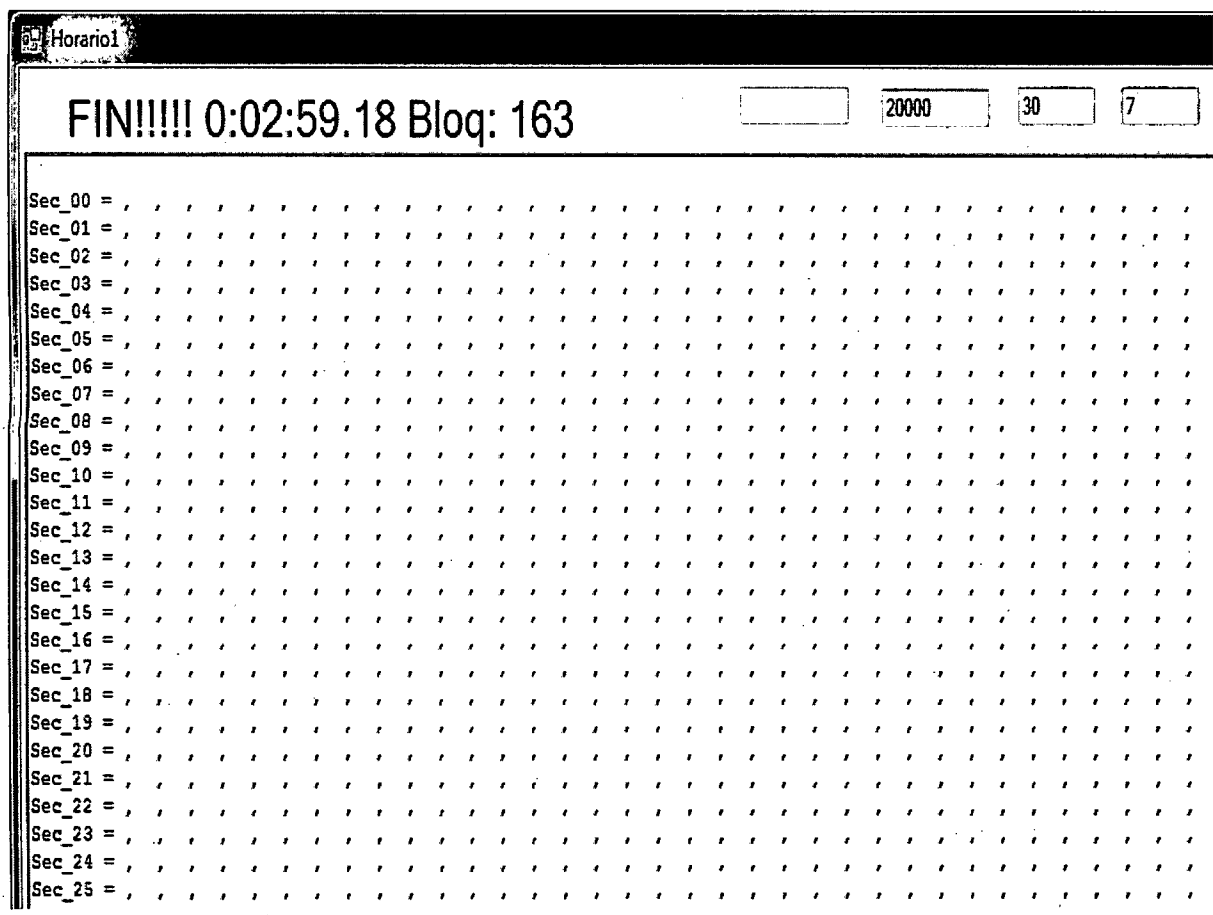
Horario1		50000		20000		30		7	
FIN!!!! 0:03:00.47 Bloq: 141									
ProC00 =
ProC01 =	.	.	.	1	.	.	.	1	.
ProC02 =	.	.	0	.	.	0	.	.	.
ProC03 =
ProC04 =	.	.	.	2	.	.	.	2	.
ProC05 =	.	.	.	2	.	.	.	2	.
ProC06 =	.	.	0	.	.	0	.	.	.
ProC07 =	.	.	0	.	.	0	.	.	.
ProC08 =	.	.	0	.	.	0	.	.	.
ProC09 =	.	.	0	.	.	0	.	.	.
ProC10 =	.	.	0	.	.	0	.	.	.
ProC11 =	.	.	0	.	.	0	.	.	.
ProC12 =	.	.	0	.	.	0	.	.	.
ProC13 =	.	.	0	.	.	0	.	.	.
ProC14 =	.	.	6	.	.	2	.	.	2
ProC15 =
ProC16 =
ProC17 =
ProC18 =
ProC19 =
ProC20 =
ProC21 =
ProC22 =
ProC23 =
ProC24 =
ProC25 =
ProC26 =	.	.	8	.	.	8	.	.	8
ProC27 =	.	.	3	.	.	3	.	.	3
ProC28 =	.	.	3	.	.	3	.	.	3
ProC29 =	.	.	4	.	.	4	.	.	4
ProC30 =	.	.	3	.	.	3	.	.	3
ProC31 =	.	.	6	.	.	4	.	.	4
ProC32 =	3	.	.	3
ProC33 =	.	.	4	.	.	4	.	.	4
ProC34 =	.	.	5	.	.	5	.	.	5
ProC35 =	.	.	5	.	.	5	.	.	5
ProC36 =	.	.	7	.	.	7	.	.	7
ProC37 =	.	.	7	.	.	7	.	.	7
ProC38 =	.	.	7	.	.	7	.	.	7
ProC39 =	.	.	8	.	.	8	.	.	8
ProC40 =	8	.	.	8
ProC41 =	.	.	11	.	.	11	.	.	11
ProC42 =	.	.	11	.	.	11	.	.	11
ProC43 =	.	.	9	.	.	9	.	.	9
ProC44 =	.	.	9	.	.	9	.	.	9
ProC45 =	.	.	8	.	.	8	.	.	8
ProC46 =	.	.	10	.	.	10	.	.	10
ProC47 =	.	.	6	.	.	10	.	.	10
ProC48 =	.	.	11	.	.	11	.	.	11
ProC49 =	.	.	1	.	.	1	.	.	1
ProC50 =
ProC51 =	.	.	9	.	.	9	.	.	9
ProC52 =

La última Memoria Tabú generada en el último ciclo de iteraciones es como sigue:

Lo cual significa que en las coordenadas dadas no se pueden asignar nuevos cambios hasta que dichas posiciones queden liberadas.

Horario1		FIN!!!! 0:03:00.47 Bloq: 141		50000	20000	30	7	PROCESA.			
Tabu00 =	.	.	759,	707,	771,	707,804,	800,759,	804,	771,	800,	
Tabu01 =	.	.	715,702,	715,697,697,	799,832,	799,686,	832,	686,	702,		
Tabu02 =	.	728,818,	818,	838,	778,838,	728,	882,	882,	778,		
Tabu03 =	.	808,	716,	820,808,820,	813,	813,	716,	813,	716,		
Tabu04 =	.	875,	797,	856,	797,	856,	875,	856,	875,		
Tabu05 =	.	869,850,	737,821,774,	869,774,792,690,850,737,	681,	681,690,	821,	792,		792,	
Tabu06 =	.	714,846,777,714,	736,	736,	877,	721,777,	846,877,	721,			
Tabu07 =	.	683,	739,	762,	747,	876,	876,	666,683,	666,739,762		
Tabu08 =	.	724,853,	724,658,	853,	853,	658,	658,	666,683,	666,739,762		
Tabu09 =	.	699,	775,	699,	870,	870,	692,775,	782,692,782,			
Tabu10 =	.	886,861,	781,815,	767,	788,	758,	781,696,861,	788,	886,717,717,815,	696,767,758	
Tabu11 =	.	.	.	701,712,	701,	712,	789,	789,			
Tabu12 =	.	802,863,	.	.	.	863,	674,	802,674,			
Tabu13 =	.	726,	763,	794,	794,	742,	763,	742,	726,		
Tabu14 =	.	.	710,753,	745,	835,710,	745,	835,753,				
Tabu15 =	.	750,	750,	848,	848,						
Tabu16 =	.	.	.	805,	749,	805,	819,	751,	749,	751,	
Tabu17 =	.	730,	684,	871,826,	871,	667,667,684,	693,	826,	693,	730	
Tabu18 =	663,	773,	773,	663,			
Tabu19 =	.	738,	874,	738,	855,	874,	731,810,	855,	679,679,	731,	810,
Tabu20 =	847,	764,	764,	847,		
Tabu21 =	.	834,	766,	814,	766,	814,	708,	865,834,708,865,			
Tabu22 =	.	798,729,729,700,867,	665,	822,	867,	754,	798,	754,665,840,	822,	840,	700,
Tabu23 =	.	.	816,816,
Tabu24 =	.	.	687,841,	885,748,	.	.	748,	841,	885,	687,	
Tabu25 =	.	845,	677,	824,	824,787,	755,	677,	787,	755,	668,845,	668,
Tabu26 =	.	695,	880,718,	.	.	695,860,	.	.	.	880,860,	
Tabu27 =	.	732,	732,785,883,	883,	839,	.	.	671,	671,785,	839,	
Tabu28 =	.	.	719,	761,	761,829,	829,	811,	719,	828,	811,828,	
Tabu29 =	.	836,866,	673,866,836,673,	.	.	
Tabu30 =	.	675,	770,	770,783,	790,	790,859,	859,	656,656,	783,	675,	
Tabu31 =	.	858,827,	744,	827,	744,	858,873,	.	.	.	873,	
Tabu32 =	.	661,	776,	806,	720,	720,	776,	806,	.	661,812,	
Tabu33 =	.	698,	694,	698,	881,	881,	676,	676,	694,		
Tabu34 =	.	837,	.	.	831,	680,660,	680,837,831,	660,			
Tabu35 =	.	757,	.	.	682,833,	682,	757,	833,			
Tabu36 =	.	711,	711,	851,	733,	784,	784,	709,851,	709,		
Tabu37 =	733,	691,	688,	688,	733,	691,	
Tabu38 =	.	817,786,	817,	807,	807,	786,	
Tabu39 =	.	727,727,	872,	854,	756,	678,854,659,678,	872,	756,	659,		
Tabu40 =	.	.	722,	722,	704,	752,	741,	864,765,	704,864,	752,741,765,	
Tabu41 =	.	.	844,844,	703,803,779,	791,	803,703,	.	.	779,	791,	
Tabu42 =	.	689,	689,	796,	796,	.	.	796,	796,		
Tabu43 =	.	830,862,	685,	830,	685,862,	884,884,	
Tabu44 =	.	.	713,	769,	713,	672,	769,	809,	825,672,	664,	825,809,
Tabu45 =	.	857,	879,725,801,	879,	725,	852,760,	662,760,	662,760,	857,852,662,	801,	
Tabu46 =	706,	670,	670,	706,		
Tabu47 =	.	878,	868,	878,	723,868,	669,	723,	843,843,669,			
Tabu48 =	.	657,	.	823,	.	657,	.	823,			
Tabu49 =	.	842,735,	705,768,	734,	768,734,	705,735,	793,842,	793,			
Tabu50 =	743,743,	.	.	.	
Tabu51 =	.	772,	780,	772,	849,	780,	740,746,740,849,746,	.	.	.	
Tabu52 =	

La Matriz de verificación de Cruces entre profesores al final quedo vacía, como debe de esperarse después de realizar el algoritmo. En este caso las 26 secciones (0....25) quedan sin ningún tipo de cruces por horas de cada profesor.



La Función de Costo por cada ciclo de Iteraciones nos genero los siguientes valores

De los cuales se puede apreciar que la Función Costo Inicial fue de 301,241 y al finalizar el calculo fue de 141, que es el menor valor posible alcanzado dados las condiciones iniciales, es decir dicha función encuentra su punto de estabilización en dicho valor.

numCosto:

Valores Iniciales:

I301241, B292249, B289253, B288253, A288252, B288252, B285251, A284252,
B27825,
...

Valores Intermedios

A3143, A3143, A3143, B3143, A3143, B3143, A3143, B3143,
A3143,
...

Y al Finalizar:

A141, A141, B141, B141, A141, A141, A141, A141,
B141, A141, A141, A141, B141, B141,

Esta corrida del Algoritmo se realizo en 3 minutos y 00.47 segundos. Y se obtuvo una función de costo final de 141, eliminándose completamente los bloqueos (cruces) entre profesores.

Los valores que empiezan con I corresponden a la evaluación Inicial de la función costo, B evaluación de la función costo después de ejecutar el núcleo que extrae bloqueos, A corresponden a la función costo después de ejecutar el núcleo aleatorio.

En el Anexo 1 se puede aprecia los valores completos que va tomando la función costo a lo largo de la ejecución de todo el algoritmo.

1.13 CONTRASTACION DE HIPOTESIS

Podemos Observar que al Desarrollar el Algoritmo mencionado, podemos encontrar una solución viable de horarios que concentre las horas de matemáticas y ciencias en la franja horaria de mayor capacidad de aprendizaje.

Además se ha realizado un estudio estadístico donde se determinó el número de horas por cada franja horaria, en los cursos de Matemáticas de 3ro y 4to de Secundaria entre los años 2008 a 2011 de un colegio determinado y se puede observar en la tabla 01 los Años, cursos, promedios por Aula, el número de horas por cada franja horaria, y el % Ponderado el cual se definió como:

$$\% \text{ Ponderado} = \frac{\left(\frac{n \text{ horas Alto}}{\text{Aprendizaje}} \right) * 3 + \left(\frac{n \text{ horas}}{\text{Aprendizaje Medio}} \right) * 2 + \left(\frac{n \text{ horas Bajo}}{\text{Aprendizaje}} \right)}{6 * (n \text{ horas Totales})}$$

AÑO	Curso	Promedio	Alta	Media	Baja	% Ponderado
2008	MAT 3º A	14.679	3	2	1	40%
2008	MAT 3º B	13.862	2	1	3	31%
2008	MAT 3º C	14.143	1	3	2	29%
2008	MAT 3º D	13.481	1	3	2	29%
2008	MAT 4º A	12.581	5	0	2	45%
2008	MAT 4º D	12.406	3	3	1	39%
2008	MAT 4º B	12.212	4	3	0	45%
2008	MAT 4º C	12.121	3	3	1	39%
2009	MAT 3º A	14.179	3	2	1	40%
2009	MAT 3º D	14.148	2	3	1	36%
2009	MAT 3º C	13.607	1	2	3	26%
2009	MAT 3º B	13.370	1	2	3	26%
2009	MAT 4º B	14.074	3	2	2	37%
2009	MAT 4º C	13.069	4	1	2	41%
2009	MAT 4º A	12.893	3	3	1	39%
2009	MAT 4º D	12.821	5	0	2	45%
2010	MAT 3º A	16.533	1	3	2	29%
2010	MAT 3º C	16.281	3	2	1	40%
2010	MAT 3º B	15.742	3	3	0	43%
2010	MAT 3º D	15.533	3	2	1	40%
2010	MAT 4º B	13.893	5	1	1	47%
2010	MAT 4º D	12.852	2	4	1	35%
2010	MAT 4º A	12.571	4	1	2	41%
2010	MAT 4º C	12.536	1	4	2	29%
2011	MAT 3º A	13.594	3	3	1	39%
2011	MAT 3º C	13.406	6	0	1	51%
2011	MAT 3º D	13.000	1	2	4	24%
2011	MAT 3º B	12.182	2	3	2	33%
2011	MAT 4º B	12.323	6	1	0	53%
2011	MAT 4º C	12.276	4	1	2	41%
2011	MAT 4º D	12.167	2	2	3	31%
2011	MAT 4º A	11.367	3	1	3	35%

Tabla 01 : Rendimiento Académico en Matemática de Alumnas de 3ro y 4to de Media

De la Tabla 01, ordenando por promedios ponderados por promoción, podemos obtener la tabla 02 y encontrar lo siguiente:

AÑO	Curso	Promedio	Alta	Media	Baja	% Ponderado		% Orden
2008	MAT 3º A	14.679	3	2	1	40%		100%
2008	MAT 3º B	13.862	2	1	3	31%		
2008	MAT 3º C	14.143	1	3	2	29%		
2008	MAT 3º D	13.481	1	3	2	29%		
2008	MAT 4º A	12.581	5	0	2	45%		75%
2008	MAT 4º D	12.406	3	3	1	39%		
2008	MAT 4º B	12.212	4	3	0	45%	Atípico	
2008	MAT 4º C	12.121	3	3	1	39%		
2009	MAT 3º A	14.179	3	2	1	40%		100%
2009	MAT 3º D	14.148	2	3	1	36%		
2009	MAT 3º C	13.607	1	2	3	26%		
2009	MAT 3º B	13.370	1	2	3	26%		
2009	MAT 4º B	14.074	3	2	2	37%	Atípico	50%
2009	MAT 4º C	13.069	4	1	2	41%		
2009	MAT 4º A	12.893	3	3	1	39%		
2009	MAT 4º D	12.821	5	0	2	45%	Atípico	
2010	MAT 3º A	16.533	1	3	2	29%	Atípico	50%
2010	MAT 3º C	16.281	3	2	1	40%		
2010	MAT 3º B	15.742	3	3	0	43%	Atípico	
2010	MAT 3º D	15.533	3	2	1	40%		
2010	MAT 4º B	13.893	5	1	1	47%		75%
2010	MAT 4º D	12.852	2	4	1	35%		
2010	MAT 4º A	12.571	4	1	2	41%	Atípico	
2010	MAT 4º C	12.536	1	4	2	29%		
2011	MAT 3º A	13.594	3	3	1	39%		50%
2011	MAT 3º C	13.406	6	0	1	51%	Atípico	
2011	MAT 3º D	13.000	1	2	4	24%		
2011	MAT 3º B	12.182	2	3	2	33%	Atípico	
2011	MAT 4º B	12.323	6	1	0	53%		75%
2011	MAT 4º C	12.276	4	1	2	41%		
2011	MAT 4º D	12.167	2	2	3	31%		
2011	MAT 4º A	11.367	3	1	3	35%	Atípico	

Promedio Total	72%
----------------	-----

Tabla 02 : Ordenamiento del Rendimiento Académico por promociones en Matemática de Alumnas de 3ro y 4to de Media

De la tabla 02 podemos observar que:

- En el Año 2008 en los salones de Tercero en Matemáticas el Orden de los promedios ponderados de aula de mayor a menor coinciden con el % de Ponderación de Horas de Mayor Aprendizaje en un 100%, mientras que para Cuarto de Secundaria el Orden coincide en un 75%.

- En el Año 2009 en los salones de Tercero en Matemáticas el Orden de los promedios ponderados de aula de mayor a menor coinciden con el % de Ponderación de Horas de Mayor Aprendizaje en un 100%, mientras que para Cuarto de Secundaria el Orden coincide en un 50%.

- En el Año 2010 en los salones de Tercero en Matemáticas el Orden de los promedios ponderados de aula de mayor a menor coinciden con el % de Ponderación de Horas de Mayor Aprendizaje en un 50%, mientras que para Cuarto de Secundaria el Orden coincide en un 75%.

En el Año 2011 en los salones de Tercero en Matemáticas el Orden de los promedios ponderados de aula de mayor a menor coinciden con el % de Ponderación de Horas de Mayor Aprendizaje en un 50%, mientras que para Cuarto de Secundaria el Orden coincide en un 75%.

Podemos realizar un promedio general de coincidencias en el Orden, para ver cual es el porcentaje de alineamientos entre el Promedio de aula y el % de Ponderación de Horas de Mayor Aprendizaje, y obtenemos un 72%. El Cual muestra la marcada influencia de las horas de mayor aprendizaje en los promedios ponderados de Aula, por lo cual se puede confirmar la hipótesis planteada.

Como se puede apreciar la relación entre el promedio por aula y el % de ponderación, a fin de asignarle similares condiciones de aprendizaje a cada sección se le debería de darles las mismas condiciones de franjas horarias,

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Al desarrollar el algoritmo se observaron algunas características propias del algoritmo tales como:

1 Al ejecutar el algoritmo existieron momentos en que el algoritmo Tabu Search caía en lazo infinito, o que no llegaba a la convergencia, por lo cual se volvía a ejecutar dicho algoritmo.

2 El algoritmo podría ser aplicado con nuevas consideraciones agregando dichos requerimientos en la Función de Costos, de tal manera que una disminución de dicha consideración represente una disminución en la función de costos.

3 En la correlación con el rendimiento académico, existieron además otras variables que inciden en el mismo, como por ejemplo la relación con el grupo humano de cada año el cual no tiene necesariamente el mismo empeño o actitud, sin embargo existe una relación entre tener más horas de cursos en las horas de mayor aprendizaje y el promedio ponderado por curso de cada aula.

4 Este algoritmo se puede extender a otras aplicaciones de carga horaria, tales como programación de procesos, asignaciones laborales, horarios universitarios, etc.

RECOMENDACIONES

1 Se recomienda poder definir adecuadamente La longitud de la Memoria Tabú, en nuestro caso al tener la matriz de profesores de 53 x El numero de periodos 35 = 1855. Se eligió una Longitud tabú de 886, como dicha memoria retiene un par de coordenadas por cada asignación tenemos 1772 que equivale a un 95% de la matriz de profesores.

Es decir solo se deja un espacio de movimientos nuevos de 5%, se llegó a esta conclusión después de muchos ensayos de prueba, observando donde existían menores casos de no convergencia, o encontrando un mejor tiempo de procesamiento.

2 Antes de cargar la información al sistema se sugiere verificar la coherencia del mismo ya que si el número de horas asignado a los profesores en una sección sobrepasa el número de periodos totales generará un caso de incongruencia y llevará al sistema a un lazo infinito.

3 Para poder lograr con mayor eficiencia se recomienda desarrollar 2 algoritmos núcleo:

El primero a través de una búsqueda aleatoria acotada por la función de costos y la Memoria Tabú.

Un segundo núcleo a través de una búsqueda de "Salones bloqueado" en el cual están asignados temporalmente a través del algoritmo mas de 1 profesor, el cual produce un desbloqueo del mismo, este mismo también esta acotado por la función de costos y la memoria Tabú.

4 Se recomienda combinar dichos núcleos anteriores para lograr una mejor convergencia, en tiempo y en calidad de la función de costos, es decir eliminar los bloqueos por cruces de profesores así como también aumentar las horas de ciencias en la franja horaria de mayor aprendizaje.

GLOSARIO DE TERMINOS

Método heurístico:

“Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.”

En contraposición a los métodos exactos que proporcionan una solución óptima del problema, los métodos heurísticos se limitan a proporcionar una buena solución del problema no necesariamente óptima. Lógicamente, el tiempo invertido por un método exacto para encontrar la solución óptima de un problema difícil, si es que existe tal método, es de un orden de magnitud muy superior al del heurístico (pudiendo llegar a ser tan grande en muchos casos, que sea inaplicable).

Tabu Search:

“Un método heurístico creado por Fred W. Glover y en el año 1986 y formalizado en 1989 Es un método de búsqueda local que se utiliza para la optimización matemática. Las búsquedas locales toman una posible solución a un problema y verifican sus vecinos inmediatos (es decir, soluciones que son similares a excepción de uno o dos detalles menores), con la esperanza de encontrar una solución mejorada. Métodos de búsqueda local tienen una

tendencia a atascarse en las regiones sub-óptimas, o en las mesetas, donde muchas soluciones igualmente aptos.

Búsqueda tabú mejora el rendimiento de estas técnicas mediante el uso de estructuras de memoria que describen las soluciones visitadas o conjuntos proporcionado por el usuario de las normas. Si una posible solución ha sido visitado previamente en un determinado período de corto plazo o si se ha violado una regla, que se marca como "tabú" ("Tabú" es una grafía distinta de la misma palabra), de modo que el algoritmo no tiene en cuenta esa posibilidad repetidamente.

BIBLIOGRAFIA

LIBROS

[L1] Daniel Borrajo, Daniel Borrajo Millán. Inteligencia artificial: métodos y técnicas. Centro de Estudios Ramón Areces, 1997.

[L2] John Haugeland. La Inteligencia Artificial. Buenos Aires Argentina, 2003.

[L3] Raúl Pino Díez, Alberto Gómez Gómez, Nicolás De Abajo Martínez. Introducción a la inteligencia artificial: Sistemas expertos, redes neuronales y computación evolutiva Universidad de Oviedo, Asturias España 2005.

[L4] Fred Glover, Manuel Laguna. Tabu Search, Volume 1 Kluwer Academic Publications Group Massachusetts USA, 2001.

TESIS

[1] Santos, Haroldo; Ochi, Luiz. A Tabu Search Heuristic with Efficient Diversification Strategies for the Class/Teacher Timetabling Problem. Ouro Preto. 2003.

<http://www.asap.cs.nott.ac.uk/patat/patat04/343.pdf>

[2] Coello Coello, Carlos. Introducción a los Algoritmos Genéticos. RED Científica.

<http://www.redcientifica.com/doc/doc199904260011.html>

[3] Terashima-Marin, Hugo; Ross, Peter; Application of the Hardness Theory when Solving the Timetabling Problem with Genetic Algorithms. Monterrey. 1999.

<http://www-csi.mty.itesm.mx/~terashim/papers/CEC-hard.ps>

<http://www-csi.mty.itesm.mx/~terashim>

[4] Gutiérrez Andrade, Miguel; delos Cobos Silva, Sergio. BUSQUEDA TABU: Un Procedimiento Heurístico para Solucionar Problemas de Optimización Combinatoria. Azcapotzalco.

<http://www.azc.uam.mx/publicaciones/enlinea2/1-3int.htm>

[5] Flores Garcia, Sudiel; Pinto Castillo, Enrique. Sistema Experto Aplicado a la Generación de Horarios Escolares. Tlaxcala. Instituto tecnológico de Tlaxcala. 2002.

<http://agentes.cic.ipn.mx/cic/cicindi2002/ponencias/sistema%20experto.PDF>

<http://personal.telefonica.terra.es/web/timetab/espanol/>

[6] Puma Villanueva, Wilfredo. Prototipo de Arquitectura de Sistema Experto de Inferencia Difusa. Arequipa.

<http://www.publicaciones.spc.org.pe/SPCMagazine/editions/I-10/html/p16-Puma.pdf>

[7] Fogel, David B. "Evolutionary Computation". Estados Unidos. IEEE Press. 1995.

[8] Ribeiro Filho, Geraldo. Tesis de Doctorado: Melhoramentos no Algoritmo Genético Construtivo e novas Aplicacoes em Problemas de Agrupamiento. São José dos Campos, Brasil. INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. 2001

[9] Silva Timóteo, Guilherme Tadeu. Desenvolvimento de um Algoritmo Genético para a Resolução do Timetabling. Lavras, Brasil. Universidade Federal de Lavras. 2002.

http://www.comp.ufla.br/curso/ano2001/Desenvolvimento_de_um_algoritmo_genetico_para_a_resolucao_do_timetabling.pdf

[10] Oliveira Costa, Eduardo; Della Bruna, Marlonn. Resolução de “Timetabling” utilizando Evolução Cooperativa. Parana, Brasil. Universidade Federal do Paraná. 2000.

<http://www.sbc.org.br/reic/edicoes/2003e1/cientificos/ResolucaoDeTimetablingUtilizandoEvolucaoCooperativa.pdf>

[11] Weare, Rupert; Burke, Edmund and Elliman, Dave. A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems. Nottingham, Gran Bretaña. University of Nottingham. 1995.

<http://www.cs.nott.ac.uk/TR/1995/1995-8.pdf>

[12] Ribeiro Filho, Geraldo, Nogueira Lorena, Luiz Antonio. A Constructive Evolutionary Approach to School Timetabling. Laboratorio Associado de Computacao e Matematica Aplicada. Brasil.

<http://www.lac.inpe.br/~lorena/geraldo/CGA-timet.pdf>

[13] Edmund Burke, Kirk Jackson; Kingston, Jeff; Weare, Rupert. Automated University Timetabling: The State of the Art. Gran Bretaña. 1999.

<http://www.asap.cs.nott.ac.uk/ASAP/publications/pdf/ekb-compj.pdf>

[14] Ender Ozcan, Alpay Alkan. Timetabling using a Steady State Genetic Algorithm. Yeditepe University. Estambul, Turquia. 2002.

<http://cse.yeditepe.edu.tr/~eozcan/research/papers/PATAT2002.pdf>

[15] Rossi-Doria, Olivia; Paechter, Ben. An hyperheuristic approach to course timetabling problem using an evolutionary algorithm. Napier University. Edimburgo, Escocia. 2002.

<http://iridia.ulb.ac.be/~meta/newsite/downloads/hyperEA.pdf>

[16] Panos M. Pardalos, Parallel Search for Combinatorial Optimization: Genetic Algorithms, Simulated Annealing, Tabu Search and GRASP. University of Florida, Gainesville. 1995.

<http://www2.research.att.com/~mgcr/doc/gqaps-parallel.pdf>

Revistas – Publicaciones

[17] López Zevallos P. El método de Búsqueda Tabú para la Programación de Horarios.

Hermosilla, Sonora.

[18] Semana.com. Tiempo de Aprender, Revista de publicación Semanal. Bogotá Colombia

http://www.semana.com/wf_ImprimirArticulo.aspx?IdArt=87757

[19] Moreno i Oliver, Francesc Xavier. Cronopedagogía. Barcelona. España.

<http://personal.telefonica.terra.es/web/fxmoreno/documen/CRONO.pdf>

ANEXOS

ANEXO 1: VALORES DE LA FUNCION COSTO

Valores de la Función Costo a través de la ejecución de todo el algoritmo.

Los valores que empiezan con I corresponden a la evaluación Inicial de la función costo, B evaluación de la función costo después de ejecutar el núcleo que extrae bloqueos, A corresponden a la función costo después de ejecutar el núcleo aleatorio.

I301241,	B292249,	B289253,	B288253,	A288252,	B288252,	B285251,	A284252,
B278252,	A278252,	B274252,	B270248,	B267252,	B267252,	B266252,	A266252,
B265252,	A265252,	B261249,	B260249,	B259249,	A259249,	B259249,	B258249,
A257251,	B253250,	B252250,	B250250,	A250250,	B246250,	B246250,	B245250,
A244250,	B238251,	B238251,	A238251,	B235247,	B234247,	A234246,	B231243,
B229243,	B228243,	B225243,	B224244,	B221244,	A221244,	B219245,	B217245,
B216242,	A215242,	A215242,	B215242,	A215242,	B209242,	A208242,	A208242,
B203243,	A203243,	A203243,	B202243,	B201243,	A201243,	B200243,	B200243,
B199243,	B199243,	B198243,	B197244,	B197244,	B197241,	B197241,	B197241,
B197241,	A197241,	B196240,	B196240,	B195240,	B194243,	A194243,	B191243,
A191239,	B189236,	A189233,	B189225,	B189220,	B189220,	B188224,	B187224,
B186224,	B184225,	B183226,	A183226,	B183226,	A183226,	B183226,	B183226,
B183226,	B182229,	B182229,	B181229,	B180229,	A180229,	B178229,	B177229,
B177229,	B176229,	B176229,	B176229,	B176229,	B176229,	B176229,	B176229,
B176229,	B176226,	B175226,	B175226,	B175226,	A175226,	B175226,	A175226,
B175226,	B171226,	B170226,	B170226,	A170226,	B170226,	B170226,	B169223,
B167223,	B167223,	B165224,	B164230,	B161230,	B160230,	B159230,	A159230,
B158230,	B156234,	B156233,	B156233,	B156233,	B156233,	B155239,	B155239,
B155239,	B154239,	B153239,	B153239,	B151236,	B151235,	B150235,	B149235,
B148235,	B148235,	B148235,	B148235,	A148235,	B148235,	A148235,	B147235,
B146235,	B145239,	A145239,	B144240,	B144240,	B143240,	B143240,	B143236,
A143236,	B142236,	B141233,	A141233,	B140229,	B140229,	B139231,	B139231,
A139231,	A139231,	B139231,	B137231,	B137231,	B137231,	A137231,	A137231,
A137228,	B136228,	A135224,	A135224,	B134226,	B132225,	B132224,	B132224,
A132224,	A132224,	A132224,	B132224,	B130220,	B129221,	A129221,	A129221,
B129217,	B129217,	B128218,	A128218,	B126218,	B126217,	B126217,	A126217,
A126217,	B126217,	B125217,	B125216,	A125216,	B123221,	A123221,	B122221,
B122218,	B122218,	B122218,	B122218,	B122218,	A122218,	B122218,	B122218,
B121218,	B120215,	A120215,	A120215,	B120215,	A120215,	B120215,	A120215,
B120215,	B120215,	B120215,	B119215,	A119215,	B118215,	B118215,	B117215,
B116216,	B115216,	B115215,	A114215,	B113215,	B113215,	A113215,	B112215,
A112215,	B112215,	B111214,	B111214,	B111214,	A111214,	B111213,	A111213,
B110213,	B110212,	B110212,	A110212,	B110212,	B110212,	B110212,	B110212,
B109212,	B109210,	B108210,	B106210,	B105210,	A105210,	B105210,	B105210,
B105210,	A104210,	B104210,	B104210,	A104210,	A104210,	B104210,	B103212,

A103212,	B103212,	B103212,	A103212,	B101216,	B101216,	B100215,	B99216,
B99216,	B99216,	B99216,	B99216,	A99216,	A99216,	B99216,	B99216,
B97216,	B97216,	A97216,	A97216,	B97216,	B97212,	B96212,	B95212,
B95212,	B94212,	B94210,	B94210,	B94209,	B94209,	B94209,	B93209,
B92209,	B90210,	B90210,	B90210,	B90210,	A90210,	B90210,	B90210,
B90210,	B89210,	B89210,	A89210,	A89210,	B89210,	B88210,	A88210,
B88210,	B87210,	A87210,	B87210,	B87210,	B87210,	B87210,	A87210,
B87210,	B87210,	B87210,	B87210,	B86210,	B86210,	B84210,	B84210,
B84210,	A84210,	B83210,	A82210,	B82210,	B82210,	B82210,	B82210,
B82210,	B82210,	B82210,	B80210,	A80210,	B80210,	A80210,	A80210,
B80210,	A80210,	B80210,	B78210,	B78207,	A78207,	A78207,	A78207,
B78207,	B77208,	A77208,	A77208,	B77208,	B77207,	B77207,	B77207,
A77207,	B77206,	A77206,	B77206,	B77206,	B76206,	B76206,	B76205,
A76205,	B76205,	B76205,	A76205,	B75205,	A75205,	A75205,	A75205,
B75205,	B75205,	A75201,	B72205,	B72205,	B72205,	A72205,	A71205,
B71205,	A71205,	A71205,	A71205,	B71205,	A71205,	B71205,	B70206,
A70206,	A70206,	B70206,	A70206,	A70206,	A70206,	A70206,	B70206,
A70206,	B70206,	A70206,	B69206,	B69206,	B69206,	A69206,	B68206,
B68206,	A68206,	B68206,	B68206,	B68206,	B67207,	B67207,	A67207,
A67207,	A67207,	A67207,	B67207,	B67203,	A67203,	A67203,	B67203,
B67199,	A67199,	B67199,	B67199,	A67199,	B66198,	B66198,	A66198,
A66198,	B66198,	B66198,	A66197,	A66197,	B66197,	A66197,	A66197,
B66197,	A66197,	B66197,	B66197,	A66197,	A66197,	B66197,	A66197,
A66197,	A66197,	B66197,	B65202,	B65202,	B65201,	B65201,	A65201,
B65201,	B65198,	A65198,	B65198,	B65198,	A65198,	B65196,	A65196,
B65196,	A65196,	B65194,	B65194,	B65194,	A65194,	A65194,	A65194,
B65193,	A65193,	B65193,	B65193,	B65193,	B65193,	B65193,	A65193,
B65193,	A65193,	B65193,	B65193,	B65193,	B64194,	B64194,	B64194,
B63198,	B63198,	B62198,	A62198,	A62198,	B62198,	A62198,	B62198,
A62198,	A62198,	A62198,	B62198,	B62198,	B62198,	B62197,	A62197,
B62197,	B61198,	A61198,	B61198,	A61198,	A61198,	A61198,	B61198,
A61198,	B61198,	B60198,	B60198,	A60198,	B60198,	B60198,	A60198,
A60198,	B60197,	B60193,	A60193,	B60193,	B60193,	B60193,	A60193,
B60193,	A60193,	B60193,	B60193,	B60193,	B60193,	B60193,	A60193,
B60193,	B60193,	B59193,	A59193,	B59193,	B59193,	A59193,	A59193,
B59193,	A59193,	A59193,	A59193,	A59193,	B59193,	B58193,	B58193,
B58193,	B58193,	A58193,	B58193,	A58193,	A58193,	A58193,	A58193,
A58193,	B58193,	B58193,	A58193,	B58193,	B58193,	A58193,	B58193,
B58193,	A58193,	B58193,	A58193,	B58193,	B58193,	B58192,	B58192,
B58192,	A58192,	A58192,	A58192,	B58192,	B58192,	A58192,	A58192,
A58192,	A58192,	A58192,	A58192,	B58192,	B58192,	B58192,	B57192,
B57192,	A57192,	B56191,	A56191,	B56191,	B56191,	A56191,	A56191,
B56191,	A56191,	B56191,	A56191,	B56191,	A56191,	A56191,	A56191,
B56191,	B55191,	A55191,	A55191,	A55191,	B55191,	A55191,	A55191,
A55191,	B55191,	A55191,	A55191,	B54191,	B54191,	A54191,	A54191,
A54191,	A54191,	B54191,	B54191,	B54191,	A54191,	B54191,	B53191,
B53191,	B53191,	B53191,	A53191,	B53191,	B52191,	B52191,	A52191,
B51191,	B51191,	B51191,	A51191,	A51191,	A51191,	A51191,	B51191,
A51191,	B51191,	B51191,	B51191,	B51191,	A51191,	B51191,	B50191,
B50191,	A50191,	A50191,	B50191,	A50191,	A50191,	A50191,	A50191,
B50191,	A50191,	A50191,	A50191,	A50191,	A50191,	A50191,	A50191,
B50191,	A50191,	B50191,	B50191,	B50191,	B50190,	A50190,	B50190,
A50190,	B50190,	A50190,	B49190,	B48190,	A48190,	A48190,	B47190,
A47190,	A47190,	A47190,	B47190,	A47190,	A47190,	B47190,	A47190,
A47190,	B47190,	A47190,	B47190,	B47190,	B46190,	B46190,	B46190,
B46190,	B46190,	A46190,	B46190,	B46190,	B46190,	A46190,	A46190,
B46190,	B46190,	A46190,	A46190,	B46190,	A46190,	A46190,	B46190,
A46190,	A46190,	A46190,	B46190,	A46190,	A46190,	B46190,	A46190,
A46190,	B46190,	A46190,	B46190,	A46190,	A46190,	B46190,	A46190,
B46190,	B46190,	A46190,	B46190,	B46190,	B46190,	B46190,	B46190,
A46190,	A46190,	A46190,	B46190,	A46190,	B46187,	B46187,	B46187,
A46187,	B46187,	B46186,	A46186,	B46186,	B46186,	B46184,	B46184,
B46184,	B46184,	B46184,	A46184,	A46184,	A46184,	A46184,	B46184,
B46184,	B46184,	B46184,	B46184,	B46184,	B46184,	B46184,	B46184,
B46184,	B46184,	A46184,	B46184,	B46184,	B46180,	B46179,	B46179,
B46179,	B46179,	B46179,	B46179,	A46179,	A46179,	B46179,	A46179,
A46179,	B46179,	B46179,	B46179,	B46179,	A46179,	B46179,	B46179,

